TECHNISCHE UNIVERSITÄT DARMSTADT

Technische Universität Darmstadt
Department of Computer Science
Chair of Applied Cryptography

Master Thesis

Trustless incoercible sale of physical goods over a blockchain

Nikolaos Stivaktakis

September 26, 2019

Supervisors: Prof. Sebastian Faust, Ph.D.
2nd supervisor

Abstract

Write an abstract

!!! Prüfen Sie, dass der folgende Text aktuell ist (entsprechend der formalen Regeln des Studienbüros) !!!
!!! Check that this text is up to date (according to formal rules of the examination office) !!!

Erklärung zur Abschlussarbeit gemäß § 22 Abs. 7 APB TU Darmstadt

Hiermit versichere ich, **Nikolaos Stivaktakis**, die vorliegende Master-Thesis / Bachelor-Thesis gemäß § 22 Abs. 7 APB der TU Darmstadt ohne Hilfe Dritter und nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die Quellen entnommen wurden, sind als solche kenntlich gemacht worden. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.
Mir ist bekannt, dass im Falle eines Plagiats (§38 Abs.2 APB) ein Täuschungsversuch vorliegt, der dazu führt, dass die Arbeit mit 5,0 bewertet und damit ein Prüfungsversuch verbraucht wird. Abschlussarbeiten dürfen nur einmal wiederholt werden.
Bei einer Thesis des Fachbereichs Architektur entspricht die eingereichte elektronische Fassung dem vorgestellten Modell und den vorgelegten Plänen.

English translation for information purposes only: Thesis Statement pursuant to § 22 paragraph 7 of APB TU Darmstadt

I herewith formally declare that I, **Nikolaos Stivaktakis**, have written the submitted thesis independently pursuant to § 22 paragraph 7 of APB TU Darmstadt. I did not use any outside support except for the quoted literature and other sources mentioned in the paper. I clearly marked and separately listed all of the literature and all of the other sources which I employed when producing this academic work, either literally or in content. This thesis has not been handed in or published before in the same or similar form.
I am aware, that in case of an attempt at deception based on plagiarism (§38 Abs. 2 APB), the thesis would be graded with 5,0 and counted as one failed examination attempt. The thesis may only be repeated once.
For a thesis of the Department of Architecture, the submitted electronic version corresponds to the presented model and the submitted architectural plans.

Datum / Date

Unterschrift / Signature

List of Figures

List of Tables

Contents

1 Introduction

[Nikos: Say in a trade without trust: who sends first? which is a big part of the dilemma.] The remote and anonymous nature of blockchains greatly complicates the sale of physical goods over a blockchain: Since the item is physical and therefore unknown to the blockchain, one cannot build a smart contract that atomically exchanges it for its price. In addition to that, the anonymity of both the seller and the buyer make it virtually impossible to build a robust recovery mechanism if one party pays/sends honestly first but the other misbehaves afterwards. The most common solution is to add a trusted third party which obtains both the coins and the good before sending them to their recipients. As third parties can misbehave and there are scenarios where no trusted third party is available, a protocol that is not relying on a trusted third party is beneficial.

The goal of this thesis is to examine whether such a protocol can exist. For that, we first have to formalize the problem. We have to define the requirements for such a protocol, what properties such a protocol has to meet in order to realize a trustless and incoercible sale. We take a game-theoretic approach to model the behaviour of the different parties: Each party is modeled rational, meaning that each party is trying to maximize its own utility/payoff/profit? We want to find a protocol where it is profitable for each party to not deviate from this protocol. If such a protocol is found, as it is in the best interest of the parties to follow the protocol, it solves the problem of a sale without trust. The seller can expect the Buyer to pay not because of trust, but because it is the most profitable option for the buyer himself. We propose various candidates and analyse if these proposed protocols do indeed meet the criteria to be such a protocol.

1.1 Motivation

In the last years there has been a big rise of the digital world and cryptocurrencies. As they gain acceptance, they become a valid payment option for sales of digital and physical goods. Current solutions work with a trusted third party, an escrow, that collects the payment and the good and then forward them to their recipients. In order for this solution to work, both parties have to trust the third party. There exist situations where there is no trusted third party and these solutions are not viable. Worse, if the two parties trust a third party that is not trustworthy, there is a possibility that they lose all the money that is involved in this trade. **[Nikos: maybe mention the big amount of scams in the ctypto world]** Therefore there is a demand for trustless solutions which solve the problem without needing a trusted third party.

While a trustless sale of digital goods over the blockchain is possible and there are existing solutions [AK19], the sale of physical goods are much more complicated as the good and its delivery are unknown to the blockchain. Finding a protocol that enables a trustless and incoercible sale of physical goods over a blockchain would make cryotocurrencies more useful as a currency and allow potential sellers to sell physical goods directy to their clients without any risk involved.

1.2 Methodology

We take an iterative approach to find a solution to our problem. We first propose the simplest possible solution and analyse if it indeed solves our problem. If yes, we are finished. If no, we add a layer of

complexity and propose a new solution. This cycle will be continued until we find a solution or we think we exhausted the most probable solutions and there is no solution to the proposed problem. If this is the case we try and proof that there is no solution to the proposed problem.

1.3 Structure of this thesis

In chapter 2 preliminaries, 3 related work...

1.4 Related work

"Solving the Buyer and Seller's Dilemma"[AK19]
In this paper Asgaonkar and Krishnamachari proposed a protocol that allows the sale of a digital good without a trusted third party. Their solution involves a Smart Contract that collects a deposit from both the selling and the buying party. If then one party misbehaves, the Smart Contract slashes the deposit of the lying party. This solution assumes the good to be digitally verifiable. Therefore if one party is not honest, the opposing honest party can report this misbehaviour to the Smart Contract and proof that the other party indeed did misbehave. The authors also provide a game-theoretic analysis of their protocol and confirm that honest behaviour is the only Nash equilibrium.

"Two Party double deposit trustless escrow in cryptographic networks and Bitcoin."[Zim]
In this work Zimbeck proposes a protocol based on bitcoin scripts that should allow the trade between two perfect strangers even if the parties themselves can not be trusted. His approach has no need for a trusted third party. Zimbeck also implemented his protocol in an open source program called BitHalo. His approach was formally and game-theoretically analysed in the paper [Big+15]

"Escrow Protocols for Cryptocurrencies"[Gol+17]
In this paper Goldfeder et al. look at the problem of buying physical goods using cryptocurrencies with a third party acting as an escrow. They formalized the escrow problem for physical goods and introduce multiple schemes with different properties.

2 Preliminaries

2.1 Game Theory

[Nikos: first generaly describe the field] Game theory studies the interaction between self-interested players [LS08a]. It uses models to analyse games. **[Nikos: a little bit of history]** It is considered that the field began in 1944 with the publication of von Nemann & Morgenstern's "The Theory of Games and Economic Behaviour" [Ras; NMR44] Then Nash... history What is a game?? Each player wants to [LS08b]

2.1.1 Games

A game consists of players who are taking actions in order to maximize their own payoff. An action is a choice that a player can make. Each possible action for a given decision is contained in an action set. Which choice the player makes is influencing how big his payoff will be. The payoff (or utility) is a numerical value and each player gets his own payoff.

Definition 1 (Game)**.** *A game is a tuple (N,A,u), where:*

- N *is a finite set of n players*
- $A = A_1 \times \cdots \times A_n$*, where* A_i *is a finite set of actions available to player i. Each vector* $a = (a_1, ..., a_n) \in A$ *is called an action profile.*
- $u_1, ..., u_n$ *where* u_i*:* $A \to \mathbb{R}$ *is a payout (or utility) function for player i.*

[LS08a]

2.1.2 Strategies

Each player faces multiple decisions in a game. At every decision point he chooses between possible actions. These decision can depend on every information that the player knows at this stage of the game. Which action a player takes in which situation can be formulated in a strategy. A possible strategy could be: If the opposing player takes action L, I will take action R, else i will take action L.

Definition 2. *Player i's strategy* s_i *is a rule that tells him which action to choose at each instant of the game, given his information set.* [Ras].

We call the set of every strategy for a player his strategy set. A strategy profile is an ordered set consisting of one strategy for each of the n players in the game. - best response -dominated and dominant strategies here

2.1.3 Nash Equilibrium

tell something about nash
what about pareto/strictly dominated strategies equilibrium?

Definition 3 (Nash Equilibrium)**.** *A strategy profile* s^* *is a Nash equilibrium if no player has incentive to deviate from his strategy given that the other players do not deviate.* [Ras] *Formally:*

$$\forall i : \forall s'_i : u(s^*_i, s^*_{-i}) \geq u(s'_i, s^*_{-i})$$

.

strict/weak nash

2.1.4 Turn-based games

Definition 4 (turn-based games)**.** *A game is turn-based if only one party can play at a time and its move is revealed to all other parties before the next move.*

To analyse turn-based games we will use game trees. The nodes of the tree represent decision points and the edges of the tree possible actions. The leafs of the tree are the payout values for the players. In this example Alice and Bob have one decision each. Alice will play first. She has the possible actions L and R. Then Bob will make his decision choosing between the actions L and R.

[Orfeas: Game trees and examples should be introduced in the preliminaries. Mention that, in our work, a game is turn-based if only one party can play at a time and its move is revealed to all other parties before the next move. Mention that we can combine simultaneous and turn-based games if some moves are turn-based and some are simultaneous.]
[Orfeas: for the example, prefer parties A, B instead of B, S. For actions, you can e.g. use L, R, which is common in Game Theory literature for generic moves when every choice is binary. Do not use the same payouts everywhere. Better use specific numbers. Use numbers with a single Nash Equilibrium and walk the reader through the analysis of why it is a Nash Equilibrium.]

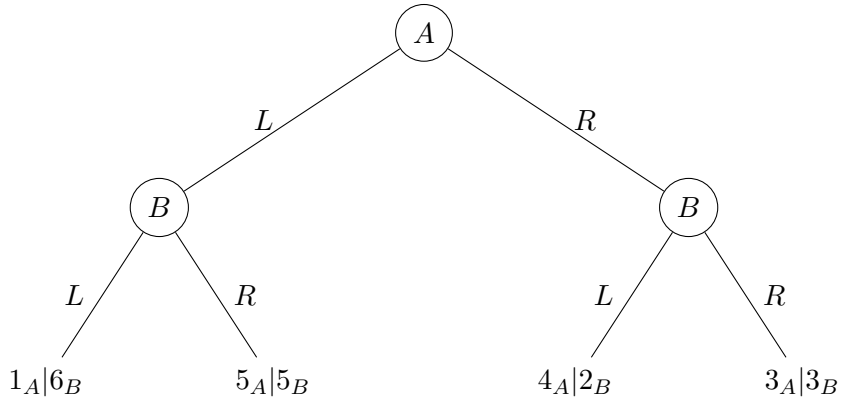


Figure 2.1: Example Tree

To find the Nash Equilibrium of this game we will use the method "iterated elimination of strictly dominated strategies". **[Nikos: TODO look for paper, more information here]**

To analyze the game we start at the bottom of the tree at Bob's decision. He always takes the action that yields him the biggest payout. We start at the left branch. If Bob takes action L, he gets a payout of 6. If he chooses action R, he gets a payout of 5. In this scenario Bob chooses L, because $6 > 5$. On the right branch he chooses R because $3 > 2$.

Then we look at Alice's decision. She knows that Bob chooses the path that yields him the greatest payout and can calculate his decisions. She knows that Bob will choose L,if Alice takes action L. Therefore her payout of action L is 1. If she chooses action R, Bob will choose R as well. In this case Alice has a payoff of 3. Alice chooses action L, because $3 > 1$.

The Nash Equilibrium of this game is $\{R,R\}$ and the payout is $3_A|3_B$.

2.1.5 Simultaneous games

Definition 5 (Simultaneous games)**.** *A game is simultaneous, if at least two parties play at a time.*

To analyse simultaneous games we will use a table. The two players are again Alice and Bob. On the top of the table we see the possible strategies of Bob: *be honest* and *lie*. On the left side are the

possible strategies for Alice: *report* and *hide*. For each combination of strategies the two parties get a specific payout. For example if Bob chooses strategy *lie* and Alice chooses strategy hide, Bob gets a payout of 8 and Alice gets a payout of 2.

Alice \ Bob	*be honest*	*lie*
report	2 | 5	5 | −3
hide	2 | 3	2 | 8

Table 2.1: Example Table

If we now look at the table we can see which combinations of strategy benefits which player. There is a Nash Equilibrium at {*be honest*, *report*}, because no player can improve their payout by changing strategy. If bob changes his his strategy to *lie*, his payout will decrease to −3. If a Alice changes her strategy to *hide*, she will get the exact same payout of 2. Because Alice's payout does not decrease, this Nash Equilibrium is a weak Nash Equilibrium.

2.2 Cryptographic primitives

[Nikos: do i need a general introduction to the field of cryptography?]

2.3 Public Key Cryptography

[Nikos: We use some PK cryptography, maybe I should explain it a little to the reader]

2.3.1 Commitment schemes

Commitment schemes are used in a vast variety of cryptographic protocols. They enable a party to commit itself to a value while keeping it secret [**Goldreich2001-bv**]. This resembles a non transparent sealed envelope: By putting a note in such an envelope, a party commits itself to the content of the note. This note can not be read until the envelope is being opened. **[Nikos: maybe delete]**

A commitment scheme lets one party (Bob) commit to a value of his choice. Later, he can open the commitment in order to convince another party (Alice) that indeed that value was committed. Committing to a value results in a commitment string c, that Bob sends to Alice, and an opening string s that Bob uses for opening the commitment later[**boneh˙graduate˙nodate**].

A commitment scheme is secure if it satisfies the following two properties:

- **Hiding:** The Commitment string c reveals no information about the commited value[**boneh˙graduate˙noda**
- **Binding:** Let c be a commitment string output that Bob gets by committing the value v. Bob can open this commitment to the committed value v, but not to any other value $v' \neq v$.

2.4 Cryptocurrencies

[Nikos: look at that again] Cryptocurrencies are decentralised currencies that introduce a way to transfer value with low latency and without any intermediary. They use various cryptographic primitives to prevent fraud. An important cryptographic tool are digital signatures, that ensure that only the owner of the coin can spend it. Hash algorithms allow For this work, it is not important to understand these cryptographic primitives. A detailed description of said crypgtographic primitives can be found at Katz and Lindell's book *Introduction into modern Cryptography* [**Katz2020-aj**].

We will explain some of the principles of cryptocurrencies by looking at The first and best known cryptocurrency: Bitcoin. Bitcoin was introduced in 2008 Satoshi Nakamoto [Nak]. In Bitcoin everyone can create his own wallet and send and receive coins. These coins are sent using transactions. Every transaction is stored on a public database, the blockchain (c.f. section 2.4.1). Once these transactions are on the blockchain, they can never be changed or deleted. The blockchain acts as a public ledger recording all the transactions. The blockchain is maintained by the miners **[Nikos: what is a miner? how deep should I explain? as miners do not have something to do with my thesis directly]**

After Bitcoin many other cryptocurrencies were developed. In this work, we do not look at a specific cryptocurrency. We rather use them as a remote way to transfer value between two parties. These two parties do not need to know each other, rather they only need each other's wallet address to send coins to each other. The transfer of value occurs nearly instantaneous and can not be reversed.

2.4.1 Blockchain

In this work we look at public blockchains used for cryptocurrencies. In this context a blockchain is a public, distributed database storing every transaction. These transactions are grouped in blocks, where every block is linked with the previous block, forming a chain of blocks. **[Nikos: maybe a figure explains it best]** New transactions are written to the blockchain in the form of new blocks that are appended to the blockchain. As new transactions are posted constantly, the blockchain is always growing. This property is called *liveliness*. Another important property is that blockchains are *immutable*, which means whatever is written on the blockchain can not be altered.
properties: immutability, liveliness
Cryptography is being used to secure these properties Digital Signatures:[**Katz2020-aj**] talk mostly about its properties

2.4.2 Smart Contracts

3 Formalization

Definition 6 (Utility Function)**.** *The utility function $U_P(G)$ describes how much utility a physical good G provides to a player P. The output of this function is a numerical value, which corresponds to the amount of coins that have equal value as $U_P(G)$. In this work, utility will always be a positive integer.*

3.1 Entities

Definition 7 (Seller)**.** *A rational entity S that owns a specific physical good G. The seller has bigger utility in owning P coins in contrast to owning G (c.f. equation 3.1) He therefore desires to exchange G against P coins.*

$$U_S(G) + P > P > U_S(G) > 0 \tag{3.1}$$

Definition 8 (Buyer)**.** *A rational entity B that owns at least P coins. The Buyer has greater utility in owning G in contrast to owning P coins (c.f. equation 3.2). He therefore desires to exchange P coins against G.*

$$U_B(G) + P > U_B(G) > P > 0 \tag{3.2}$$

[Orfeas: The SC is not really an escrow, as it doesn't hold on to the physical good, only the money] [Orfeas: This definition is not detailed enough. You may introduce the exact SC code here (discussing collateral/handling of P in case of disagreement as well), or at least refer to it TODO: rethink code location] [Nikos: I am a little unsure, how I want to define SC]

Definition 9 (Smart Contract)**.** *An entity SC that should ensure that the trade is successful. It acts as a fully trusted oracle.*

3.2 The Game

[Nikos: is this the right headline??]

Definition 10 (Initial State)**.** ***[Nikos: look at that definition again]*** *A Seller and a Buyer have some way of communicating with each other. A successful trade is beneficial for both entities. Both parties have enough funds in the given cryptocurrency to carry out protocol actions. The Seller is in possession of the physical good.*

Definition 11 (Desired Final State)**.** *The Buyer is in possession of the physical good and his funds have decreased by amount P. The funds of the seller have increased by P.*

Definition 12 (Success of the trade)**.** *The trade is considered successful if the situation transitions from the Initial State to the Desired Final State.*

We make the following assumptions:

1. All entities have access to the Smart Contract and know the addresses and public keys of each other.

2. The seller knows the physical address to which the buyer wants the good to be shipped. This address is not necessarily the buyer's real physical address, he can choose any physical address he wants to protect his privacy.

3. Other than the information of assumption 1 and 2, the buyer and the seller know nothing about each other.

4. Both parties can send funds in the given cryptocurrency to any address.

5. The sender has the power to send/ship the physical good to any physical address he wants.

6. Both the seller and the buyer know the code of the Smart Contract.

7. The seller and the buyer behave rationally, meaning that they both want to maximise their expected payout.

8. Both $U_S(G)$ and $U_B(G)$ are positive.

We do not analyse transaction fees.

3.2.1 Goal

The goal of this work is to design a protocol in which behaviour that results in a successful trade is a Nash Equilibrium. Minimal trust should be required. This will be achieved, if it is the rational behaviour for both players to perform the trade. A successful trade should be a Nash Equilibrium: If the other party behaves honestly, the best strategy will be to behave honestly as well. **[Nikos: Repitition? redundant?]**

Another goal is that the trade is incoercible against rational attacks. That means that it should be unprofitable attack and decrease the payout of the other player, if the other player is honest. If this goal is achieved, a it is unprofitable, a rational Player will never try and attack the opposing player. It should be noted that that an irrational player can attack the opposing party, as an attack is unprofitable, but not necessarily impossible. If a player is willing to lower his own utility, he can attack an opposing honest player.
[Orfeas: The "Success of the trade" seems a bit too mixed with the "goal". Try to express it in a clear way, maybe even merging the two concepts.] **[Nikos: now okay? I set success of the trade as a pure definition. Goal is in text form and describes a little more]**

4 Candidate Two-Party Protocols

In this chapter we try to find a two-party protocol that solves the proposed problem. We start by a simple, naive approach and then increase the complexity on each proposed protocol. On each protocol we examine the created game and analyze the Nash Equilibrium of that game. As described in section 3.2, we want to create a game where the Nash Equilibrium lies in an honest trade between the Seller and the Buyer.

4.1 First try: no collateral

The simplest protocol is both players sending their resources without any collateral nor a third party. In this case there are 3 possibilities:

Buyer sends first	
1 :	Buyer sends P coins to the Seller
2 :	Seller sends the physical good G to the Buyer

Figure 4.1: Protocol: No collateral, Buyer sends first

Seller sends first	
1 :	Seller sends the physical good G to the Buyer
2 :	Buyer sends P coins to the Seller

Figure 4.2: Protocol: No collateral, Seller sends first

Both players send simultaneously	
1 :	Buyer sends P coins to the Seller and seller sends G to the Buyer simultaneously

Figure 4.3: Protocol: No collateral, both players send at simultaneously

4.1.1 Possible strategies

There is a countless amount of possible strategies for the two players. For example, a player can follow the protocol and trade honestly, he can try and gain an advantage by not following the protocol, he can ignore all messages or he can send all his money to the other party. It is not sensible to list and analyse all the possible strategies. We will concentrate on the strategies where a player follows the protocol and the strategies where a player intentionally deviates from the protocol to gain an advantage. As soon as we find a strategy for a player that is more profitable than following the protocol, we know that this protocol is not achieving our goals, as a rational player will not follow it. If we conclude that following the protocol is the most profitable strategy for both players, we know that this protocol is achieving our goal. **[Nikos: look at the end of this paragraph again]**

For the protocols proposed in section 4.1 we analyse two strategies for each player. The Buyer can choose between strategy *Pay* where he behaves honestly and sends the money and strategy *Withhold*

where the Buyer behaves dishonestly and does not send the money. The Seller can choose as well between strategy *Send* where he sends the good G and strategy *Keep* where the Buyer does not send the G.

4.1.2 Game: The Buyer acts first

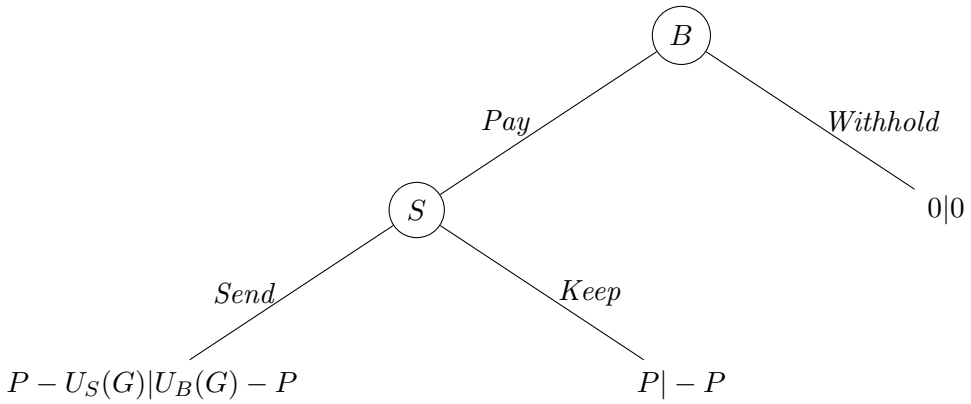


Figure 4.4: No collateral, Buyer acts first

To analyze the game, we use the method "iterated elimination of strictly dominated strategies" **[Nikos: refer to preliminaries/source]** . We start at the bottom at the seller's decision.

- The Seller chooses *Keep* (Because $P > P - U_S(G)$). **[Nikos: Do you like this format with itemize? I want the decisions to visually stand out]**

We then go one step towards the root and look at the Buyer's decision. He already knows that if he chooses *Pay*, the rational Seller will choose *Keep*. This would leave the Buyer with a payout of $-P$. If the Buyer chooses *Withhold*, his payout is 0.

- The Buyer chooses *Withhold* (Because $0 > -P$)

The Nash Equilibrium of this game is {*Keep*, *Withhold*} and results in a payout of 0|0. Both parties are not sending anything and the trade is unsuccessful.

4.1.3 Game: The Seller acts first

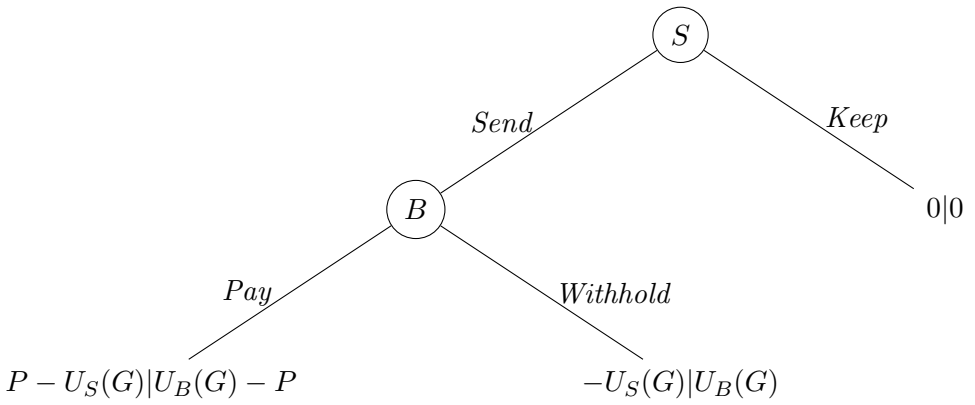


Figure 4.5: No collateral, Seller acts first

The analysis is similar to the above case, but the roles are reversed.

- The Buyer would choose *Withhold* as $U_B(G) > U_B(G) - P$

- The seller would choose *Keep* as $0 > -U_S(G)$

The Nash Equilibrium of this game is {*Keep*, *Withhold*} and results in a payout of $0|0$. Again, both parties are not sending anything and the trade is unsuccessful.

4.1.4 Game: Both parties act simultaneously

	Pay	*Withhold*
Send	$P - U_S(G) \mid U_B(G) - P$	$-U_S(G) \mid U_B(G)$
Keep	$P \mid -P$	$0 \mid 0$

Table 4.1: No collateral: Both parties send simultaneously

If we look at the payout table we can see that *Withhold* dominates *Pay* and *Keep* dominates *Send*. These strategies provide a bigger payout regardless of the strategy of the other party. Since we assume the players to be rational, they would both choose the dominant strategy. Again, the Nash Equilibrium of this game is {*Keep*, *Withhold*} and results in a payout of $0|0$. As the Nash Equilibrium does not correspond to the Desired Final State (c.f. Definition 11), this game does not satisfy our goal.

It is interesting to note that this table resembles the famous prisoner's dilemma, where the Nash Equilibrium is not Pareto efficient. We can see that {*Keep*, *Withhold*} is not the Pareto optimal case, because {*Send*, *Pay*} increases the payout of both parties. In other words, a successful trade would benefit both parties but is not happening in this game as it is not the Nash Equilibrium. **[Nikos: last sentence redundant?]**

4.2 Second Try: Buyer Collateral

To incentivize an entity to follow the protocol we introduce collateral. The collateral will be lost if the entity deviates from the protocol. In this section we will look at a protocol where only one the Buyer deposits collateral. We introduce a Smart Contract (SC) which handles the collateral. The payment P is now routed via the Smart Contract.

Following protocol is proposed:

Buyer Collateral protocol

1 : Buyer deposits collateral $C > P$ to the Smart Contract
2 : Seller sends the physical good G to the seller
3 : After receiving G, the Buyer sends funds of amount P to the Smart Contract
4 : The Smart Contract releases the collateral of the Buyer and sends P to the Seller

Figure 4.6: Buyer Collateral protocol

The Smart Contract releases the collateral as soon as it receives P by the Buyer. By paying P, the Buyer implicitly reports to the Smart Contract that he has received the physical good G. The collateral should incentivize the Buyer to pay the Seller after receiving the physical good G, as his collateral is greater than P.

It is important that the Seller sends first, as he has no collateral and therefore no incentive to follow the protocol. If we swap the order and let the Buyer send first, the Seller will simply keep G and be in possession of both G and P. Because of the collateral, the Seller can safely send G first without having to worry that the Buyer will not pay him.

4.2.1 Possible Strategies

We still have the strategies from section 4.1.1: *Pay* and *Withhold* for the Buyer and *Keep* and *Send* for the Seller. The The Buyer now has one additional decision: He can either follow the protocol and deposit the collateral C (*Deposit*) or deviate from the protocol and refuse to deposit the collateral (*Refuse*).

4.2.2 Game

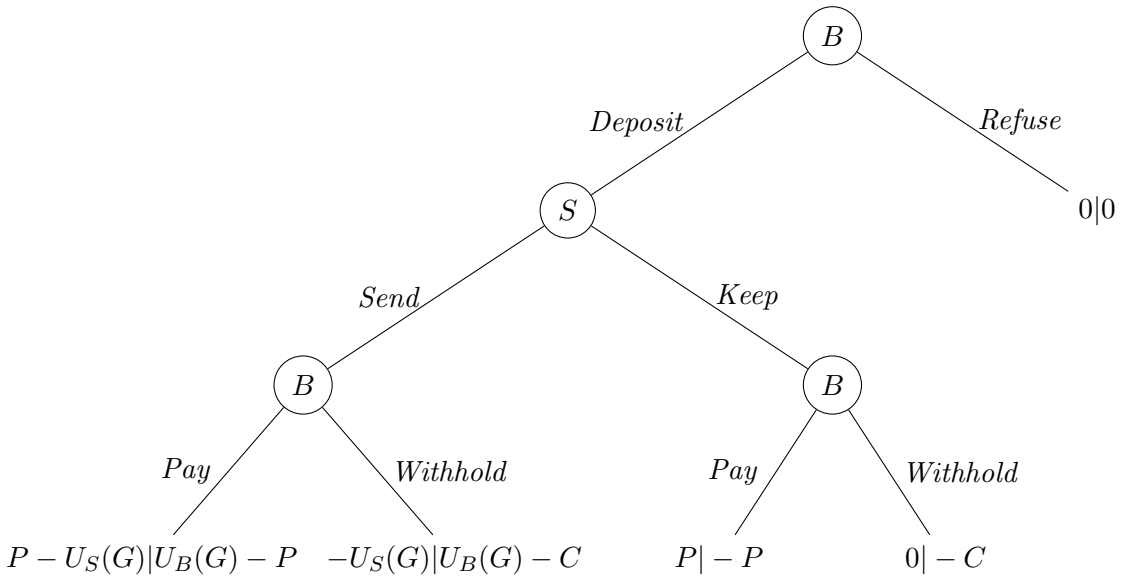


Figure 4.7: Only the Buyer deposits collateral

To analyze the payout of this game, we again start at the bottom at the Buyer's decision. On both branches the Buyer chooses *Pay*:

- $U_B(G) - P > U_B(G) - C$
- $-P > -C$

We now look at the Seller's decision. He chooses *Keep*:

- $P > P - U_S(G)$

This means that it is better for the Seller to behave maliciously and not send the physical good G. The Buyer would send P just to get back his collateral but receive nothing more, leaving him with a payout of $-P$. At the root of the tree the Buyer chooses $Refuse$:

- $-P < 0$

The Nash Equilibrium is at $\{Refuse\}$ and results in a payout of $0|0$. The trade is unsuccessful.

4.3 Third Try: Seller Collateral

In this section we will look into protocols where only the Seller deposits collateral.

Following protocol is proposed:

Seller collateral protocol	
1 :	Seller deposits collateral $C > P$ to the Smart Contract
2 :	Buyer sends funds of amount P to the seller
3 :	Seller sends the physical good to the Buyer
4 :	Buyer receives the physical good and confirms it to the Smart Contract
5 :	The Smart Contract releases the collateral of the Seller

Figure 4.8: Seller collateral protocol

The collateral should incentivize the Seller to behave honestly and to ship the physical good to the buyer. The Buyer can report to the SC with a simple boolean input:

- 0 denotes that G was not received
- 1 denotes that G was received

4.3.1 Possible Strategies

There are still the basic strategies from section 4.1.1: *Pay* and *Withhold* for the Buyer and *Keep* and *Send* for the Seller. Similar to the strategies of the Buyer Collateral protocol (c.f. 4.2.1) the Seller has the option to deposit the collateral (*Deposit*) or refuse to deposit the collateral (*Refuse*).

The Buyer has a unique opportunity: He decides what happens to the Seller's collateral as he can either report that he received the physical good G (*Report*) or he can report that he did not receive it. Because he has this power over the Seller's collateral, he can try and coerce the Seller into sending him extra resources. This strategy will be denoted as *Coerce*. After receiving the physical good he communicates to the Seller: "I will only report that I received the good if you send me back my payment P". It is important to see that this is a credible threat, because the Buyer's payout is not affected by him reporting or not reporting. Still,he has to find a way to convince the Seller that this statement is true. For the coercion to work, the Seller has to be convinced that he will indeed receive back his collateral if he sends the Payment P back to the Seller. One possible way of convincing the Seller is via a second Smart Contract set up by the Buyer. As soon as the payment P is sent back to the Buyer, this second SC will automatically report 1 to the first Smart Contract and the Seller's collateral is released. If the Buyer chooses *Coerce*, the Seller has two possible actions: He can comply and send back P (*Comply*) or not comply and ignore the message (*Ignore*).

4.3.2 Game

To analyze the payout, we again start at the bottom. Since $C > P$, The Seller would choose *Comply* on both instances:

- $-U_S(G) > P - C - U_S(G)$
- $0 > P - C$

One step above, the Buyer chooses *Coerce* on both instances:

- $U_B(G) > U_B(G) - P$

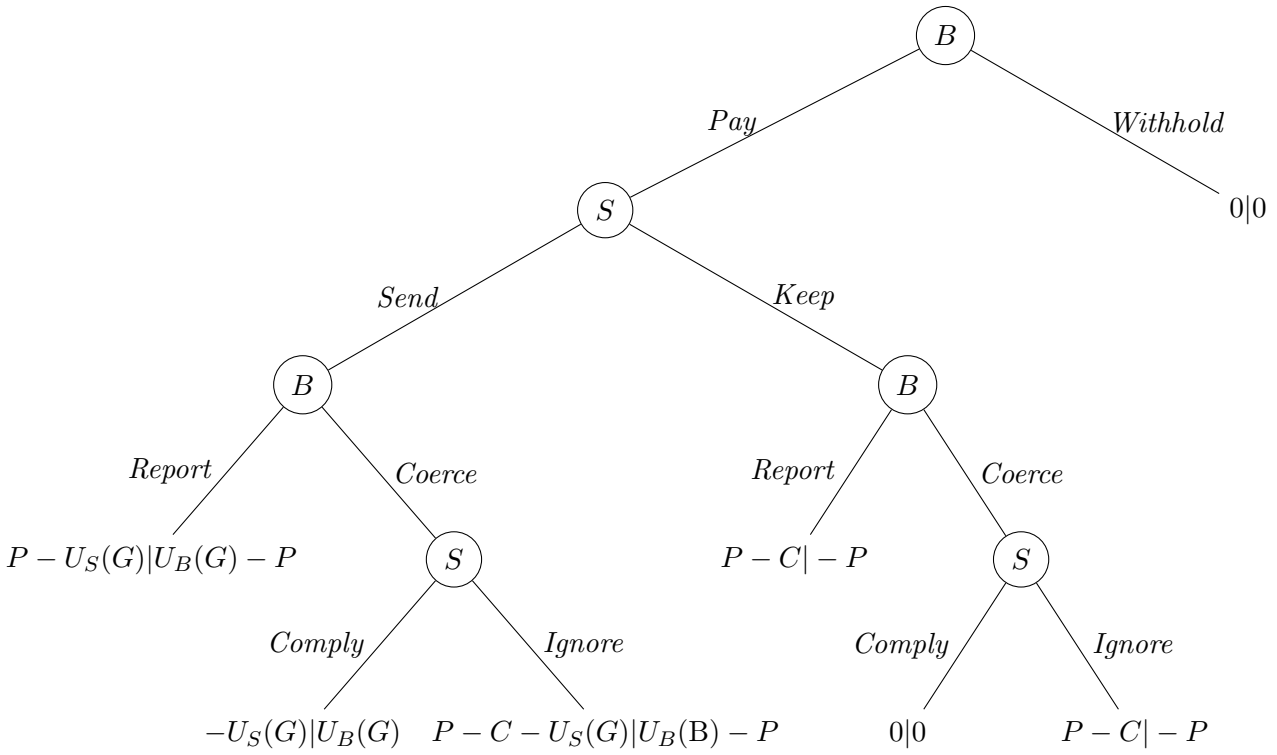


Figure 4.9: Only the Seller deposits collateral

- $0 > -P$

Based on this information the Seller chooses *Keep*:

- $0 > -U_S(G)$

That means that both options for the Buyer would result in a payout of 0|0. It does not matter which option he chooses. The Nash Equilibrium is at {Withhold} **[Nikos: am I describing the Nash Equilibria correctly?]** and results again at a payout of 0|0. The trade not successful. What was designed to be an incentive for S to behave honestly, ended up as leverage for B over S. B can report 0 (and burn the collateral) even if S behaves honestly.

4.4 Fourth try: Seller collateral, both parties report

One Side Collateral on the Seller's side does not work if only the Buyer reports to the Smart Contract (c.f. 4.3). If only the Buyer reports, he has all the power and can report anything he wants to the Smart Contract. Another option would be that both players report to the Smart Contract if the physical good G was sent and received. Also, instead of sending the Payment P directly to the Seller, the Buyer will send the payment to the Smart Contract. **[Orfeas: Isn't the payment always routed through the smart contract anyway?]** The Smart Contract will then, if both parties behave honestly, forward the payment to the Seller. This hands more power to the Smart Contract. With control over the collateral and the payment it has more options to punish dishonest behaviour. If the two players report a different value, then the Smart Contract will slash the collateral, as one party has to be lying,whereas if the parties agree, their collateral is returned. If they both report 0 (G was not sent), then the Smart Contract will send P back to the Buyer, because G was not shipped. If they both report 1 (G was sent), then the Smart Contract will send P to the Seller and the trade will be successful.

The Following protocol is proposed:
There are multiple ways to implement step 4: Either the two parties report one after another or they report simultaneously.

Seller collateral protocol: Both parties report	
1 :	Seller deposits collateral $C > P$ to the Smart Contract
2 :	Buyer sends funds of amount P to the Smart Contract
3 :	Seller sends the physical good to the Buyer and the Buyer receives it
4 :	Buyer and Seller report the physical transaction to the Smart Contract
5 :	The Smart Contract releases the collateral of the Seller and sends P to the Seller

Figure 4.10: Seller collateral protocol: Both parties report

4.4.1 Possible Strategies

There are still the same strategies as in section 4.3.1. We now denote reporting 0 to the SC as "0" and reporting 1 to the SC as "1".

4.4.2 Game: The Seller reports first

As the Buyer reports last, this protocol still gives the Buyer the power of burning the collateral of the Seller by reporting exactly the opposite from the Seller's report. Therefore he can still coerce the Seller into sending him back his payment. As we already have shown in section 4.3, a rational seller will always comply and send back the payment P. This last decision from the Seller (leaf 3, 6, 9, 12) will be omitted from the tree for better readability).

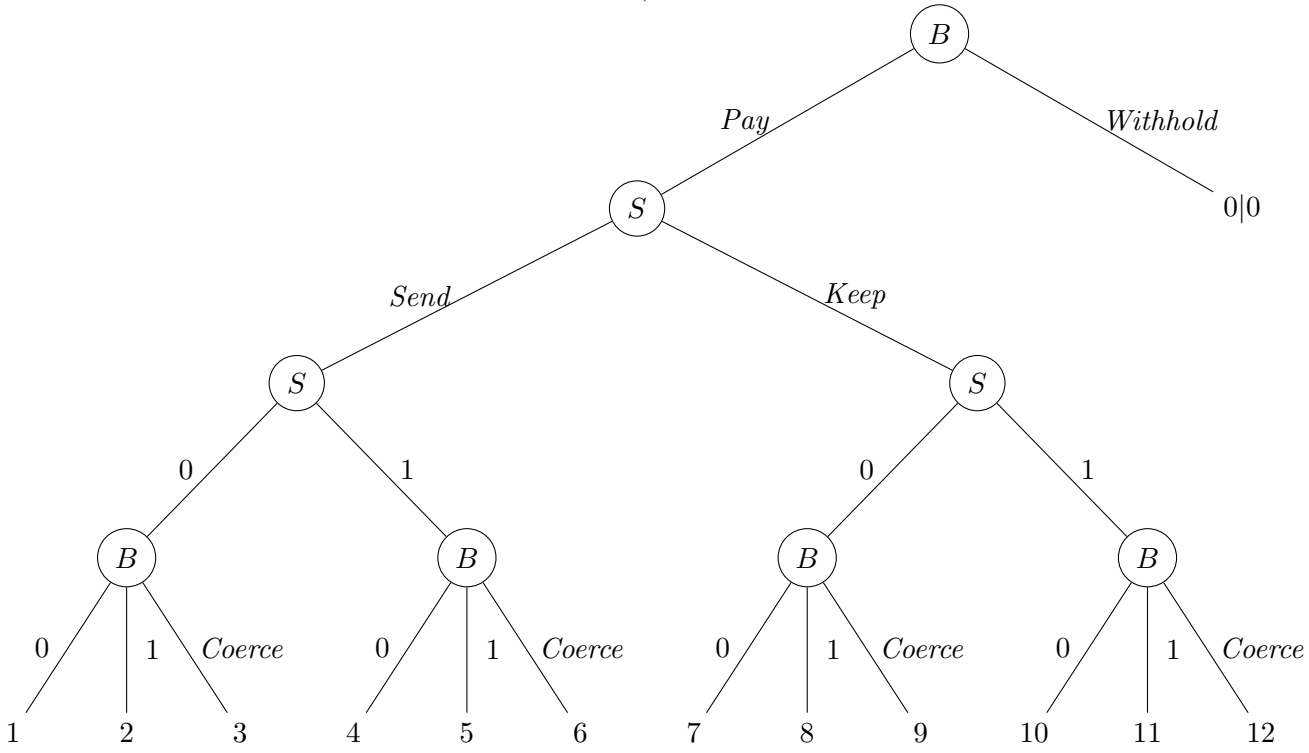


Figure 4.11: Only the Seller deposits collateral: the Buyer reports first

To analyze the payout, we again start at the bottom. The Buyer chooses:

- 1 or 3 over 2 ($U_B(G) > U_B(G) - P$)
- 6 over 5 and 4 ($U_B(G) > U_B(G) - P$)
- 7 or 9 over 8 ($0 > -C$)
- 12 over 10 and 11 ($0 > -P$)

	Payout
1	$-U_S(G)\|U_B(G)$
2	$-C-U_S(G)\|U_B(G)-P$
3	$-U_S(G)\|U_B(G)$
4	$-C-U_S(G)\|U_B(G)-P$
5	$P-U_S(G)\|U_B(G)-P$
6	$-U_S(G)\|U_B(G)$
7	$0\|0$
8	$-C\|-P$
9	$0\|0$
10	$-C\|-P$
11	$P\|-P$
12	$0\|0$

Table 4.2: The corresponding payout table of figure 5.6

The Buyer chooses:

- 7 or 12 over 1 and 6 ($0 > -U_S(G)$)

No matter which decision the Buyer takes at the top of the tree, the payout will be $0|0$ and the trade unsuccessful.

4.4.3 Game: The Buyer reports first

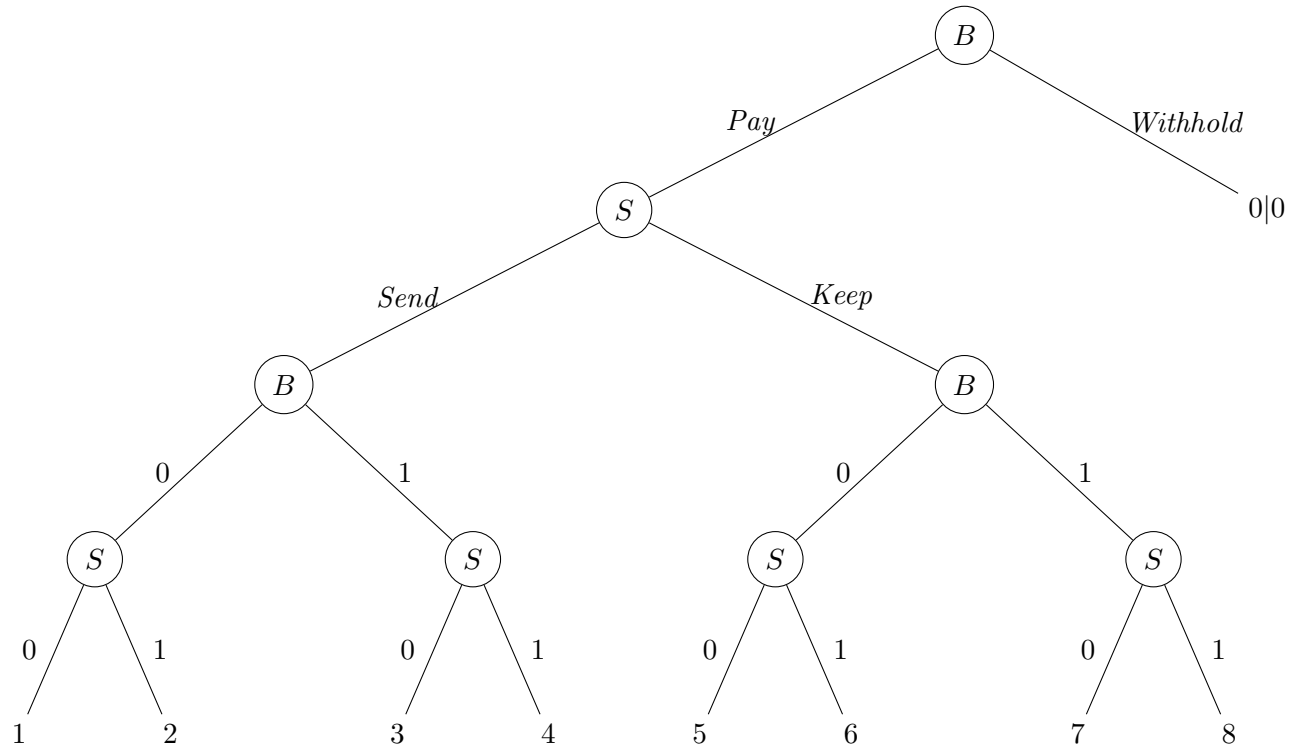


Figure 4.12: Only the Seller deposits collateral: the Buyer reports first

To analyze the payout, we again start at the bottom. The seller chooses:

- 1 over 2 ($-U_S(G) > -C-U_S(G)$)
- 4 over 3 ($P-U_S(G) > -C-U_S(G)$)

	Payout
1	$-U_S(G) \mid U_B(G)$
2	$-C - U_S(G) \mid U_B(G) - P$
3	$-C - U_S(G) \mid U_B(G) - P$
4	$P - U_S(G) \mid U_B(G) - P$
5	$0 \mid 0$
6	$-C \mid -P$
7	$-C \mid -P$
8	$P \mid -P$

Table 4.3: The corresponding payout table of figure 5.7

- 5 over 6 ($0 > -C$)
- 8 over 7 ($P > -C$)

The Seller always reports the same as the Buyer, because he will lose his collateral if he reports anything different.
The Buyer chooses:

- 1 over 4 ($U_B(G) > U_B(G) - P$)
- 5 over 8 ($0 > -P$)

The Buyer always reports that he did not receive the item. Based on this information the Seller chooses *Keep*:

- $0 > -U_S(G)$

No matter which decision the Buyer takes at the top of the tree, the payout will be $0|0$ and the trade unsuccessful.

4.4.4 Game: Both parties report simultaneously

How to report simultaneously

Every transaction to the Smart Contract is public. As the players have to report to the Smart Contract by sending a public transaction, the other player also knows what the first player reported. By just demanding both to report at the same time, the player who reports second always knows what value the first player reported. Then the report is not really simultaneous.

We can fix this problem by using a computationally hiding and binding commitment scheme. Both players would commit to their decision and send the commitment to the SC. The second player sees the commitment of the first player. Because the commitment scheme is computationally hiding and we assume the players to be computationally bounded, the commitment does not reveal any information about the actual decision of the first player. After both players send the commitment to the SC, they both send the opening value for their commitment to the SC. As the commitment is computationally binding, the players cannot change what they initially reported. If a player does not send an opening value after a predefined time, the SC punishes the player by sending all the funds to the other player.

T_1	0_B	1_B
0_S	$-U_S(G) \mid U_B(G)$	$-U_S(G) - C \mid U_B(G) - P$
1_S	$-U_S(G) - C \mid U_B(G) - P$	$P - U_S(G) \mid U_B(G) - P$

Table 4.4: Subgame T1: Both players report simultaneously

The Nash Equilibrium of subgame T_1 is $\{0_B, 0_S\}$ and payout $-U_S(G) \mid U_B(G)$.

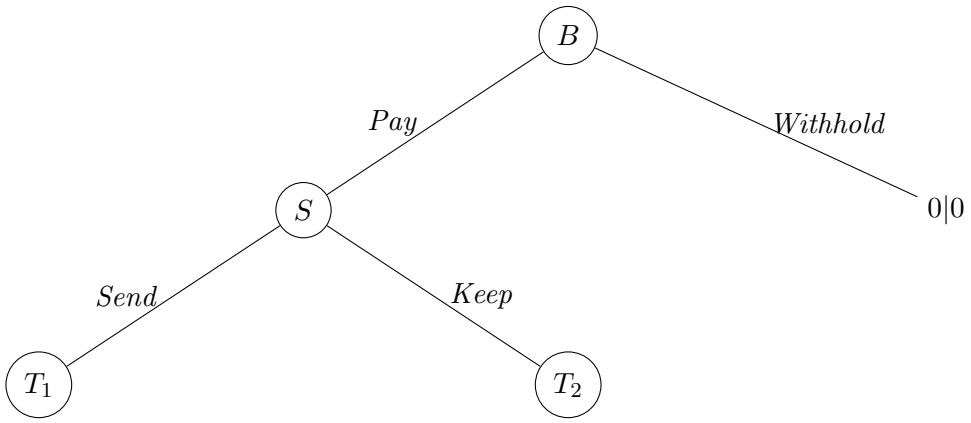


Figure 4.13: Only the Seller deposits collateral: Both players report simultaneously

T_2	0_B	1_B
0_S	$0\|0$	$-C\|-P$
1_S	$-C\|-P$	$P\|-P$

Table 4.5: Subgame T2: Both players report simultaneously

The Nash Equilibrium of subgame T_2 is $\{0_B, 0_S\}$ and payout $0|0$.

S would choose *Keep*, because $0 > -U_S(G)$. As both options lead to a payout of 0, B chooses an arbitrary option. The trade is unsuccessful again.

4.5 Fifth Try: Two side collateral

As we saw in the previous two sections, one side collateral does not lead to a game that achieves our goal. The party that does not deposit collateral always has an unfair advantage over the other party. When only the Buyer deposits collateral, the Seller has no incentive to send G. When only the Seller deposits collateral, the Buyer can coerce the Seller into sending back his payment P, because the Buyer can burn the collateral by reporting 0.

We introduce another protocol that addresses these problems. In this protocol both parties deposit collateral. This should incentivize the parties to both behave honestly. If the two parties report something different, the Smart Contract will slash the collateral of both players, as it can not determine which party is lying. If both parties report 0, the Smart Contract will send the payment P back to the Buyer. If both parties report 1, the Smart Contract will send the payment P to the Seller.

Again, there are three possible ways of handling the reports:

- The Seller reports first
- The Buyer reports first
- Both players report simultaneously

4.5.1 Possible strategies

There are still the basic strategies from section 4.1.1: *Pay* and *Withhold* for the Buyer and *Keep* and *Send* for the Seller. In addition to that the players can either report 0 or 1 to the SC, similar to section 4.4.1.

4.5.2 Game: The Seller reports first

The following protocol is proposed:

Two side collateral: The seller reports first	
1 :	Seller deposits collateral $C_S > P$ to the Smart Contract
2 :	Buyer deposits collateral $C_B > P$ and P to the Smart Contract
3 :	Seller sends physical good G to the Buyer and reports it to the Smart Contract
4 :	Buyer receives the physical good and reports it to the Smart Contract
5 :	The Smart Contract releases the collateral of the Seller and the Buyer and sends P to the Seller

Figure 4.14: Two side collateral protocol: Seller reports first

To increase readability, we do not plot the decisions from point 1 and point 2 of the protocol. If someone does not deposit the collateral, the Smart Contract aborts the trade and the payout is $0|0$.

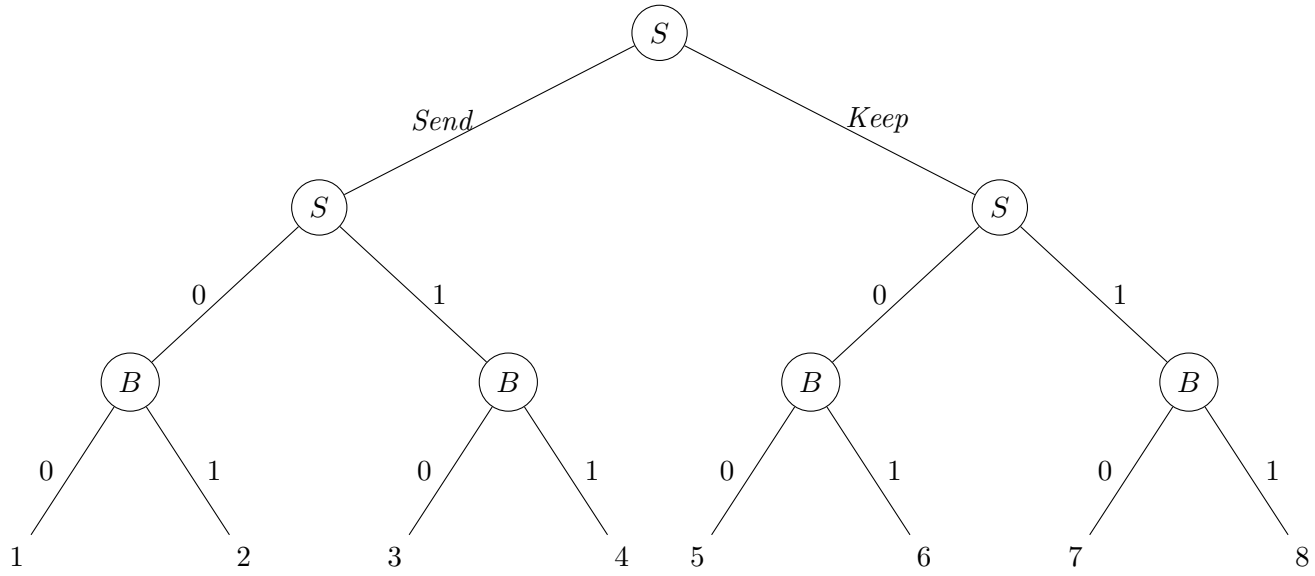


Figure 4.15: Two Side Collateral: The Seller reports first

	Payout
1	$-U_S(G)\|U_B(G)$
2	$-U_S(G) - C_S\|U_B(G) - P - C_B$
3	$-U_S(G) - C_S\|U_B(G) - P - C_B$
4	$P - U_S(G)\|U_B(G) - P$
5	$0\|0$
6	$-C_S\| - P - C_B$
7	$-C_S\| - P - C_B$
8	$P\| - P$

Table 4.6: The corresponding payout table of figure 5.9

To analyze the payout, we start at the bottom of the tree. The Buyer would choose:

- 1 over 2 ($U_B(G) > U_B(G) - P - C_B$)
- 4 over 3 ($U_B(G) - P > U_B(G) - P - C_B$)

- 5 over 6 ($0 > -P - C_B$)
- 8 over 7 ($-P > -P - C_B$))

The Seller would choose option 8:

$$P > P - U_S(G) > 0 > -U_S(G)$$

Because the payout of the Buyer is $-P < 0$, he would not engage in this trade and therefore refuse to pay the collateral in step 1. The trade is still unsuccessful.

4.5.3 Game: The Buyer reports first

The following protocol is proposed:

Two side collateral: The Buyer reports first	
1 :	Seller deposits collateral $C_S > P$ to the Smart Contract
2 :	Buyer deposits collateral $C_B > P$ and price P to the Smart Contract
3 :	Seller sends physical good G to the Buyer
4 :	Buyer receives the physical good and reports it to the Smart Contract
5 :	Seller reports to the Smart Contract that he did send G
6 :	The Smart Contract releases the collateral of the Seller and the Buyer and sends P to the Seller

Figure 4.16: Two side collateral protocol: Buyer reports first

To increase readability, we again do not plot the decisions from point 1 and point 2 of the protocol.

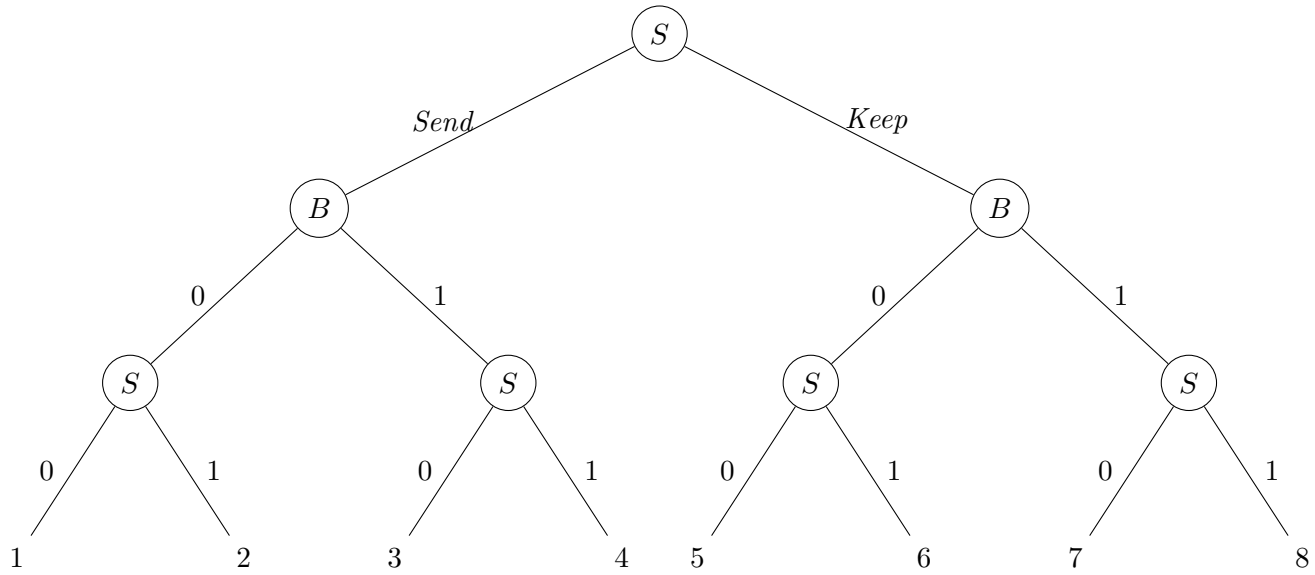


Figure 4.17: Two Side Collateral: The Buyer reports first

To analyze the payout, we start at the bottom of the tree. The Seller would choose:

- 1 over 2 ($-U_S(G) > -U_S(G) - C_S$)
- 4 over 3 ($P - U_S(G) > -U_S(G) - C_B$)
- 5 over 6 ($0 > -C_S$)
- 8 over 7 ($P > -C_S$)

	Payout
1	$-U_S(G)\|U_B(G)$
2	$-U_S(G) - C_S\|U_B(G) - P - C_B$
3	$-U_S(G) - C_S\|U_B(G) - P - C_B$
4	$P - U_S(G)\|U_B(G) - P$
5	$0\|0$
6	$-C_S\| - P - C_B$
7	$-C_S\| - P - C_B$
8	$P\| - P$

Table 4.7: The corresponding payout table of figure 4.17

The Buyer would choose:

- 1 over 4 ($U_B(G) > U_B(G) - P$)
- 5 over 8 ($0 > -P$)

The Buyer would always report 0, no matter if the Seller was honest or not. The Seller chooses *Keep*, as $0 > -U_S(G)$. The Nash Equilibrium of this game is {*Keep*, 0,0} and the payout is 0|0. The trade is unsuccessful.

4.5.4 Game: Simultaneous report

The following protocol is proposed:

Two side collateral: Simultaneous report	
1 :	Seller deposits collateral $C_S > P$ to the Smart Contract
2 :	Buyer deposits collateral $C_B > P$ and price P to the Smart Contract
3 :	Seller sends physical good G to the Buyer
4 :	Both players report whether G was shipped
5 :	The Smart Contract releases the collateral of the Seller and the Buyer and sends P to the Seller

Figure 4.18: Two side collateral protocol: Simultaneous report

Again, we do not plot the decisions from point 1 and point w. If someone does not deposit the collateral, the Smart Contract aborts the trade and the payout is 0|0.

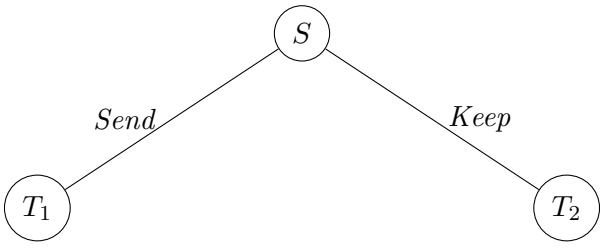


Figure 4.19: Two Side Collateral: Both parties report simultaneously

The subgame T_1 has two Nash Equilibria: $\{0_B, 0_S\}$ and $\{1_B, 1_S\}$.
The payout of $\{0_B, 0_S\}$ is $-U_S(G)|U_B(G)$. The payout of $\{1_B, 1_S\}$ is $P - U_S(G)|U_B(G) - P$.
The subgame T_2 has the same two Nash Equilibria as well: $\{0_B, 0_S\}$ and $\{1_B, 1_S\}$.
The payout of $\{0_B, 0_S\}$ is 0|0. The payout of $\{1_B, 1_S\}$ is $P| - P$.

T_1	0_B	1_B
0_S	$-U_S(G) \mid U_B(G)$	$-U_S(G) - C_S \mid U_B(G) - P - C_B$
1_S	$-U_S(G) - C_S \mid U_B(G) - P - C_B$	$P - U_S(G) \mid U_B(G) - P$

Table 4.8: Subgame T1: Both parties report simultaneously

T_2	0_B	1_B
0_S	$0 \mid 0$	$-C_S \mid - P - C_B$
1_S	$-C_S \mid - P - C_B$	$P \mid - P$

Table 4.9: Subgame T2: Both parties report simultaneously

The Seller decides if he wants to enter subgame T_1 or T_2. To analyse his decision we look at his payout from each subgame:

1. T_1
   - $\{0_B, 0_S\} : -U_S(G)$
   - $\{1_B, 1_S\} : P - U_S(G)$
2. T_2
   - $\{0_B, 0_S\} : 0$
   - $\{1_B, 1_S\} : P$

If we compare these 4 possible payouts:

$$P > P - U_S(G) > 0 > -U_S(G)$$

Without any additional assumptions, it is unclear which is the better subgame for the Seller. He can not choose which subgame is beneficial for him.
We have created a situation where the players do not have all the information they need to choose the best option. That also means, that they are not motivated to behave honestly as they do not know if this is indeed the most beneficial behaviour. As the players are not motivated to behave honestly, they would not trade successfully in this protocol.

4.6 An extra input is needed

The Smart Contract plays an important role in the two side collateral protocol. We want to create a Smart Contract that rewards honest behaviour and punishes dishonest behaviour.
Problems arise, because the Smart Contract does not know if the physical good G was actually sent to the Buyer. Because of this missing information, it is impossible for the Smart Contract to settle disputes.

4.6.1 Ideal Functionality

We can achieve the ideal functionality of our Smart Contract by giving it the information whether the good was actually shipped as an input. This ideal functionality is unrealizable without some trusted party that always reports truthfully to the SC. We model the ideal behaviour with a function taking 6 inputs and giving 2 outputs.

The irrational case (line 7) should never appear in the real world. At least one player is lying and thereby risking his collateral. Either the Seller did send G but reports that he did not, or B did not

```
1  function assign_coins_ideal(
2    buyerCol: natural, sellerCol: natural, price: natural,
3    sellerSent: bool /* self reported */, buyerRecv: bool /* self reported */,
4    shipped: bool /* ground truth */
5  ) -> (int /* seller coins */, int /* buyer coins */) {
6    if sellerSent == buyerRecv { /* honest case, multiple possible solutions  */ }
7    if (not sellerSent) and buyerRecv { /* irrational case, multiple possible solutions
       */ }
8
9    // both claim they went through and the other player failed
10   if shipped {
11     sellerCoins: int >= price + sellerCol // honest Seller
12     buyerCoins: int < price + buyerCol - U_B(G) // lying Buyer
13   } else {
14     sellerCoins: int < sellerCol // lying Seller
15     buyerCoins: int > price + buyerCol // honest Buyer
16   }
17   return (sellerCoins, buyerCoins)
18 }
```

Figure 4.20: General Code for the ideal functionality

receive G but reports that he did. Both behaviours are irrational as the lie would harm the lying party. The irrational case can be implemented without looking at the ground truth. The same can be said about the honest case, as there is no dispute to settle.

The important case is when the seller reports that he shipped the good and the Buyer reports that he did not receive it. Because the Smart Contract has the input 'shipped', it can determine which party is lying and which party is telling the truth. The Smart Contract settles the disputes in such a way that the honest party has a payout greater than zero and the lying party has a payout smaller than zero. Therefore it is unfavorable for any party to report wrongfully.

We will now see why the function outputs are defined in this way to achieve the ideal functionality:

Honest Seller
The honest Seller (line 11) ships the good to the Buyer and deposits collateral C_S. To have a positive payout, his lowest possible return from the smart Contract is $P + C_S$. His payout is at least

$$P + C_S - C_S - U_S(G) = P - U_S(G) > 0$$

Lying Buyer
The Lying Buyer (line 12) receives G and deposits collateral C_S and Price P to the Smart Contract. To have a negative payout, his return from the smart Contract has to be less than $P + Col_B - U_B(G)$. His payout is less than

$$P + C_B - U_B(G) + U_B(G) - C_B - P = 0$$

Dishonest Seller
The dishonest Seller deposits collateral C_S but does not ship the good G. To have a negative payout, his return from the smart Contract has to be less than C_S. His payout is less than

$$C_S - C_S = 0$$

Honest Buyer
The honest Buyer deposits collateral C_B and Price P. To have a positive payout, his return from the smart Contract has to be greater than $P + C_S$. His payout is greater than

$$P + C_S - C_S - P = 0$$

4.6.2 Ideal Smart Contract solves the proposed problem

In the following section we will show that an implementation of the ideal functionality of the Smart Contract indeed creates a game where a successful trade is the Nash Equilibrium. We first write an easy implementation of the ideal SC and then determine the Nash Equilibrium of the created game.

Implementation of ideal Smart Contract

The easiest implementation of the honest case is to give back their collateral. If the good was actually shipped, the Seller will be awarded coins of amount P, otherwise those coins go back to the Buyer. In the irrational case the SC will not give any coins back. In the case of a dispute the SC rewards all the coins to the honest party.

```
function assign_coins(
  buyerCol: natural, sellerCol: natural, price: natural,
  sellerSent: bool, buyerRecv: bool, shipped: bool
) -> (int /* seller coins */, int /* buyer coins */) {
  if sellerSent == buyerRecv {
    sellerCoins = sellerCol // give back the collateral
    buyerCoins = buyerCol // give back the collateral
    if shipped{sellerCoins += price} // G was shipped, Seller gets price
    else {buyerCoins += price}// G was not shipped, Buyer gets price
    }
  if (not sellerSent) and buyerRecv {
    sellerCoins = 0 // irrational case, give both parties no coins
    buyerCoins = 0
  }

  // both claim they went through and the other player failed
  if shipped {
    sellerCoins = price + sellerCol + buyerCol // honest Seller
    buyerCoins =  0 // lying Buyer
  } else {
    sellerCoins = 0 // lying Seller
    buyerCoins = price + sellerCol + buyerCol // honest Buyer
  }
  return (sellerCoins, buyerCoins)
}
```

Figure 4.21: Example Code for an implementation of the ideal Smart Contract

With this ideal Smart Contract we can create a new game. We can use any two-side collateral protocol of section 4.5. In this example we will let the Seller report first (c.f. figure 4.14).

	Payout
1	$-U_S(G) \vert U_B(G)$
2	$-U_S(G) - C_S \vert U_B(G) - P - C_B$
3	$P - U_S(G) + C_B \vert U_B(G) - P - C_B$
4	$P - U_S(G) \vert U_B(G) - P$
5	$0 \vert 0$
6	$P + C_B \vert - P - C_B$
7	$-C_S \vert C_S$
8	$P \vert - P$

Table 4.10: The corresponding payout table of figure 4.22

Remark: If any party does not send the collateral, the Smart Contract will release the collateral of the the other party and it would result in a payout of $0|0$. This option is not analyzed in the Game

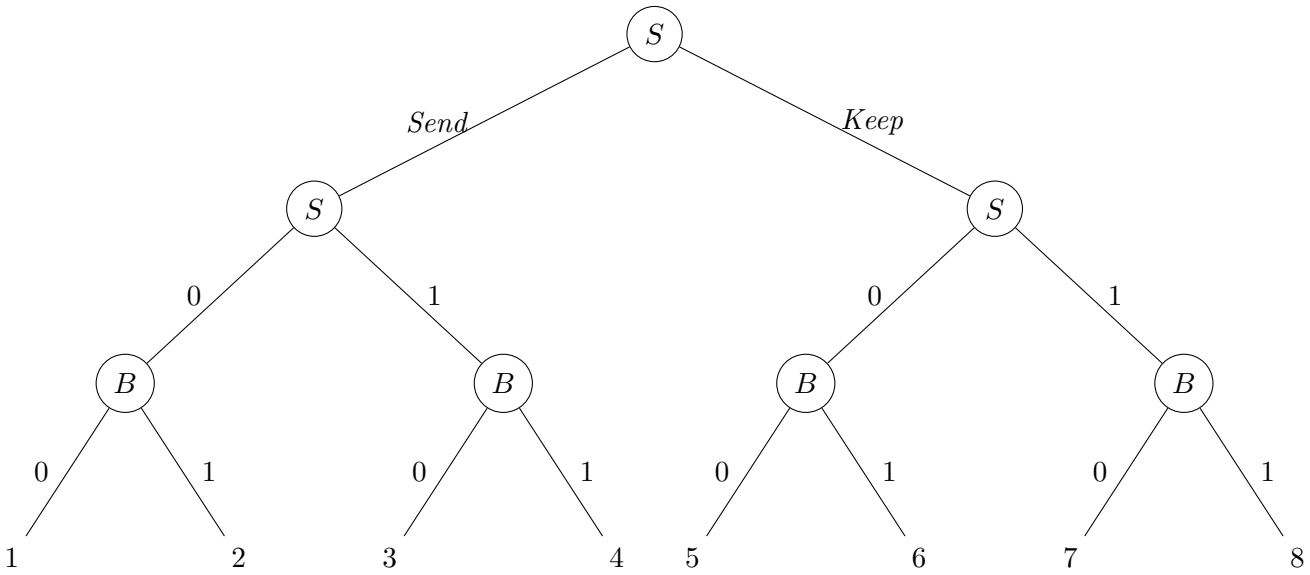


Figure 4.22: Payout with an ideal Smart Contract

tree but has to be considered as an option for each party.

We again start at the bottom of the tree at the Buyer's decision. A rational Buyer would choose:

- 1 over 2 ($U_B(G) > U_B(G) - P - C_B$)
- 4 over 3 ($U_B(G) - P > U_B(G) - P - C_B$)
- 5 over 6 ($0 > -P - C_B$)
- 7 over 8 ($C_S > -P$)

Out of these options the Seller would choose outcome 4 (*Send*,1), as this is the best outcome for him. The Nash Equilibrium is at {*Send*, 1, 1} and the trade would unravel successfully.

Such a smart contract solves the problem completely as the Nash Equilibrium lies in both parties behaving honestly.

4.6.3 Real World Functionality

[Nikos: I was working on that while we talked, so NOT finished] In the real world, the Smart Contract does not know the ground truth. Therefore the signature of a function realizable in the real world is: This poses the following problem: The ground truth is needed to settle disputes. Even if all the other

```
function assign_coins_real(
  buyerCol: natural, sellerCol: natural, price: natural,
  sellerSent: bool, buyerRecv: bool
) -> (int /* seller coins */, int /* buyer coins */)
```

Figure 4.23: Signature of a function realizable in the real world

inputs are the same, the output of the ideal functionality depends on the last input. Without this input, previously different scenarios become indistinguishable for the Smart Contract.

Theorem: There is no function $assign\_coins\_real'$ :: $(5inputs) -> (2outputs)$, such that for all $\forall bc, sc, p \in \mathbb{N}, \forall ss, bs, sh \in \{0, 1\}$,

$assign\_coins\_real$(buyercol = bc sellercol = sc, price = p, sellersent = ss, buyerRecv = bs) = $assign\_coins\_ideal$(buyercol=bc sellercol=sc, price=p, sellersent=ss, buyerRecv=bs ,shipped=s))

with = we mean that assign_coins' returns a value that is in the ranges defined by the ideal functionality.

Proof: We look at the case where $sellercol = 5, Buyercol = 5, price = 4, sellersent = 1buyerRecv = 0, shipped = 1$. By definition of the ideal functionality (c.f. figure 4.22), its payoff to the Buyer has to be:

$$assign_coins_ideal \tag{4.1}$$

For example, it is impossible to differentiate between a lying seller that did not send the good G but says he did, and a lying Buyer that received the good but says he did not. Without the sixth input, it is impossible to design a Smart Contract that achieves the ideal functionality described in section 5.4.1. **[Orfeas: do not hardcode section numbers, use labels/references instead]** **[Orfeas: I would really like to present this argument in a theorem-proof format]**

5 A redesigned game

As we have shown (c.f. **??**) **[Orfeas: we have to discuss what will be the fate of the proof]** , it is not possible (under our assumptions) to design a protocol that would create a successful game (as described in 3.2). The biggest problem is that we have no reliable way of telling the Smart Contract whether the physical good G was actually shipped or not. If there is a dispute, the Smart Contract does not know which party is lying and has to be punished.

We need to rework our assumptions and requirements for the game we want to design. For that it is is interesting to see that the two side collateral protocol (c.f. 4.5) is very close to achieving our goals. As already discussed, the problem is that the Seller is not actually incentivized to send the physical good G, because the Buyer is not incentivized to report his dishonest behaviour. The Buyer gets a bigger Payout when he lies to the Smart Contract instead of reporting the dishonest behaviour. Therefore a rational Buyer would never punish a dishonest Seller in this scenario. If a Seller thinks that the Buyer would punish him (the Buyer is not acting rational in this scenario), then the Seller is incentivized to send the physical good G.

5.1 Two Side Collateral with truthful Buyer

In this section we show that if the Buyer behaves truthful (and therefore not rationally), it is not beneficial for the Seller to behave dishonestly. The created game would have a successful trade as Nash Equilibrium.

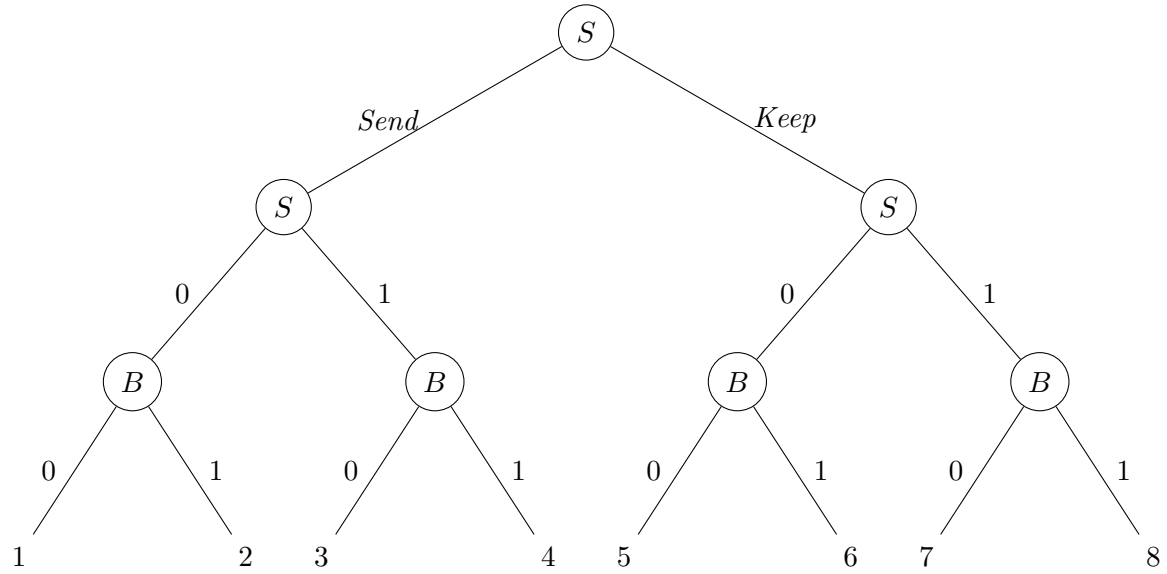


Figure 5.1: Two Side Collateral with truthful Buyer

	Payout
1	$-U_S(G) \mid U_B(G)$
2	$-U_S(G) - C_S \mid U_B(G) - P - C_B$
3	$P - U_S(G) - C_S \mid U_B(G) - P - C_B$
4	$P - U_S(G) \mid U_B(G) - P$
5	$0 \mid 0$
6	$-C_S \mid - P - C_B$
7	$-C_S \mid - P - C_B$
8	$P \mid - P$

Table 5.1: The corresponding payout table of figure 5.1

Now a truthful Buyer stops acting rationally and chooses option 7 over 8.
The Buyer chooses:

- 1 over 2 ($U_B(G) > U_B(G) - P - C_B$)
- 4 over 3 ($U_B(G) - P > U_B(G) - P - C_B$)
- 5 over 6 ($0 > -P - C_B$)
- 7 over 8 (irrational decision)

The Seller would choose option 4:

$$P - U_S(G) > 0 > -U_S(G) > -C_S$$

The Seller now sends the good, because he the Buyer will punish him if he does not send it. The Nash Equilibrium would be the honest case and the trade would unravel successfully.

At first glance this seems counterintuitive: If the act of reporting benefits the Buyer, why would the rational Buyer not report the misbehaviour of the Seller? As the definition of a rational behaviour is maximizing the own Utility, would it not be rational to report the Seller? The answer to this question is no. If the Seller has already misbehaved, it is more profitable for the Buyer to not report the misbehaviour, as he at least receives back his collateral. It is not the act of reporting that is profitable for the Buyer, rather it is the threat of reporting. This thread has to be considered by the Seller before he chooses whether to misbehave. If the Seller thinks that maybe the Buyer will report him, he is discouraged of trying to cheat and not sending G.

The problem lies in the way we laid the rules for the game: As we defined both players as always behaving rationally, the Seller knows that the Buyer can not report him, because it is not the rational decision. If the Seller did not know that the Buyer is always acting rational, he would be taking a great risk by not sending the physical good: the risk of being exposed and losing his collateral and his money P. We want to create a situation where the Seller evaluates the situation as being too risky to cheat. In this situation the rational Seller will behave honestly and send G. To conclude: It is sufficient to turn the non credible threat of the Buyer (reporting 0 if G is not shipped) into a credible threat for the double collateral protocol to succeed.

We redesign the game by adding another Entity. Its purpose is to assist the Buyer in convincing the Seller that he will be punished if he does not send G. It turns the non credible threat of the Seller's misbehaviour being reported into a credible threat. We need this Entity to maintain the rationality of the Buyer, as a rational Buyer will not report any misbehaviour of the Seller. If this is achieved, a Seller knows that it is unprofitable to not send G.

5.2 Formal definition of the new Entity

[Nikos: Maybe change title to "formally redesigning the game"] We introduce another entity, which we call Mediator M: and has the following behaviour:

Definition 13 (Mediator)**.** *An Entity M that has the purpose to protect the Buyer from an misbehaving Seller. M has the following behaviour:*

1. *receive SC address and specifications that describe G*
2. *receive the physical good*
3. *check if the the physical good match the description for G*
   - *if so: report 1 to the SC and send the good to the Buyer*
   - *else: destroy the good and report 0 to the SC*

The new Entity is not modeled as a rational party. It always behaves in the described way. It has its own physical address and the power to receive physical items. Given the right specifications of G, the Mediator can distinguish G from every other physical item. The Mediator also has the power to communicate with the Smart Contract and the Buyer. It is trusted by the Buyer to evaluate if the good sent by the Seller is indeed G. //TODO Maybe expand on that: What about trust

5.3 Mediated protocol

We propose the following protocol:

Mediated protocol	
1 :	Buyer gives SC address and specifications for G to the Mediator
2 :	Seller deposits collateral $C_S > P$ to the Smart Contract
3 :	Buyer deposits collateral $C_B > P$ and price P to the Smart Contract
4 :	Seller sends physical good G to the Mediator and reports it to the Smart Contract
5 :	Mediator receives the physical good, confirms it to the Smart Contract and sends G to the Buyer
6 :	The Smart Contract releases the collateral of the Seller and the Buyer and sends P to the seller

Figure 5.2: Mediated protocol description

5.3.1 Honest behaviour

[Nikos: Maybe honest behaviour as well with Cryptocode] This new protocol does change the honest behaviour of the Buyer. His honest behaviour is now:

1. Tell the right SC address and the right specifications for G to the Mediator
2. Deposit $C_B > P$ to the Smart Contract
3. Send funds of amount P to the Smart Contract
4. Wait and receive G from the Mediator

It is important that the Buyer no longer reports to the Smart Contract, as this responsibility is handed over to the Mediator.
The honest behaviour of the Seller remains unchanged by the new protocol. It still is:

1. Deposit $C_S > P$ to the Smart Contract
2. Send G to the provided address (which is now the Mediator's)
3. Report to the Smart Contract that G was sent

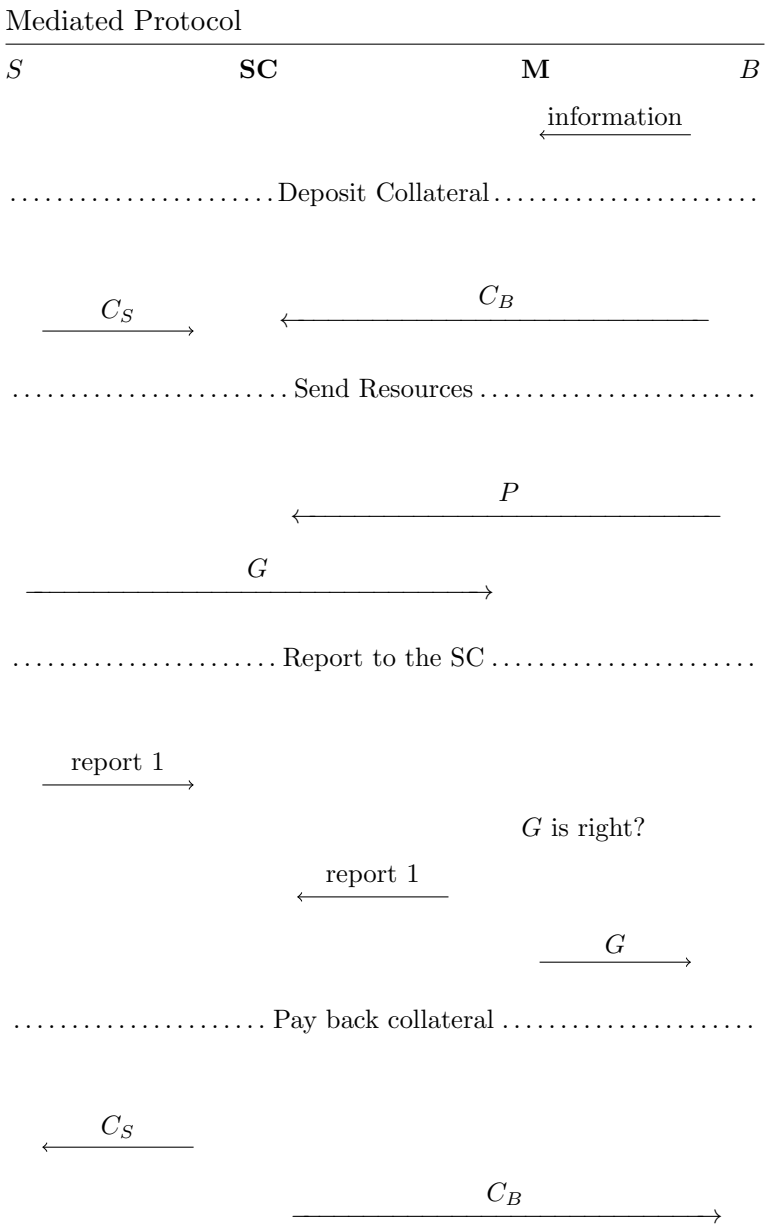


Figure 5.3: Mediated Protocol

5.3.2 Possible Strategies

The new protocol has no impact on the possible strategies for the Seller. He still has the option to either send (*Send*) or keep (*Keep*) the physical good. After that, he has the option to report 0 or 1 to the Smart Contract. As the Buyer does not longer report to the Smart Contract, he has fewer decisions to make. As soon as he enters the trade (pays the collateral and P to the SC), he no longer has any decision to make. The Mediator will take care of reporting to the Smart Contract.
We again start our analysis at point 4 of the protocol.

[Orfeas: Why does S report before M? Discuss. Mention that, at the end of the day, since this game works and we achieved our goal, we don't really care about small differences like this.] **[Nikos: maybe here complete tree as it is our final solution?]** The Seller can choose the one option which yields him the biggest payout. This is sending the good (*Send*) and reporting 1, as:

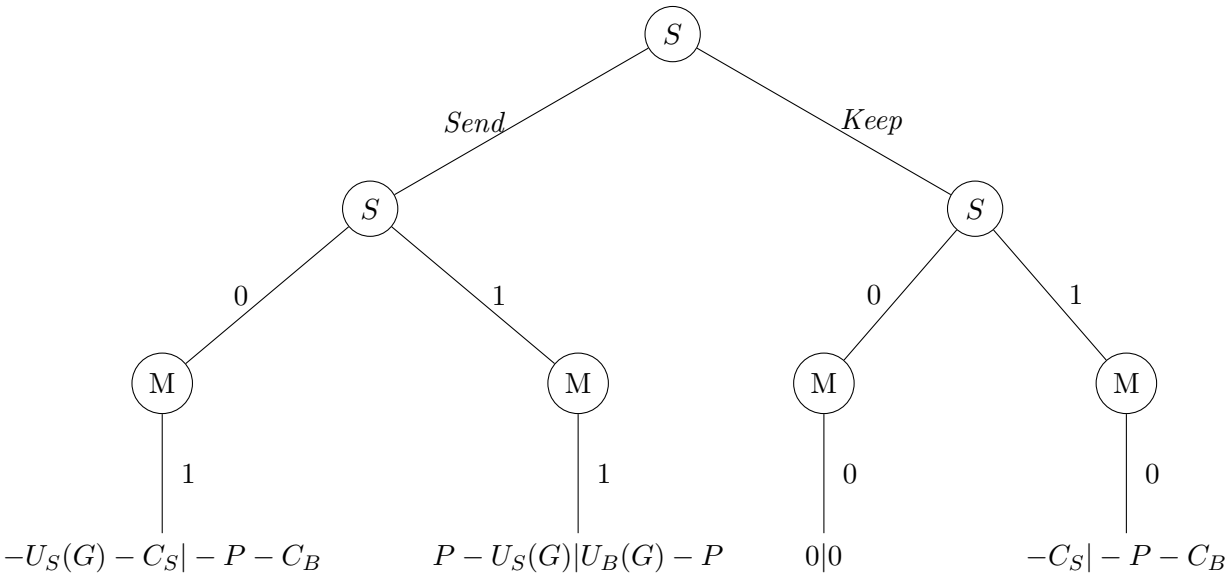


Figure 5.4: Two Side Collateral with ideal Mediator

$$P - U_S(G) > 0 > -C_S > -U_S(G) - C_S$$

It is the best option for the Seller to successfully trade with the Buyer. The Buyer would also engage in the trade, because it is profitable for him:

- $U_B(G) - P > 0$

The Nash Equilibrium of this game is both players trading successfully. This game achieves the goal formulated in section 3.2.1.

5.4 Possible realizations of Mediator

The Mediator, as it is defined in 6.1, is a theoretical Entity. In the following section we will discuss some practical attempts to realize the Mediator.

5.4.1 Friend of Buyer reports

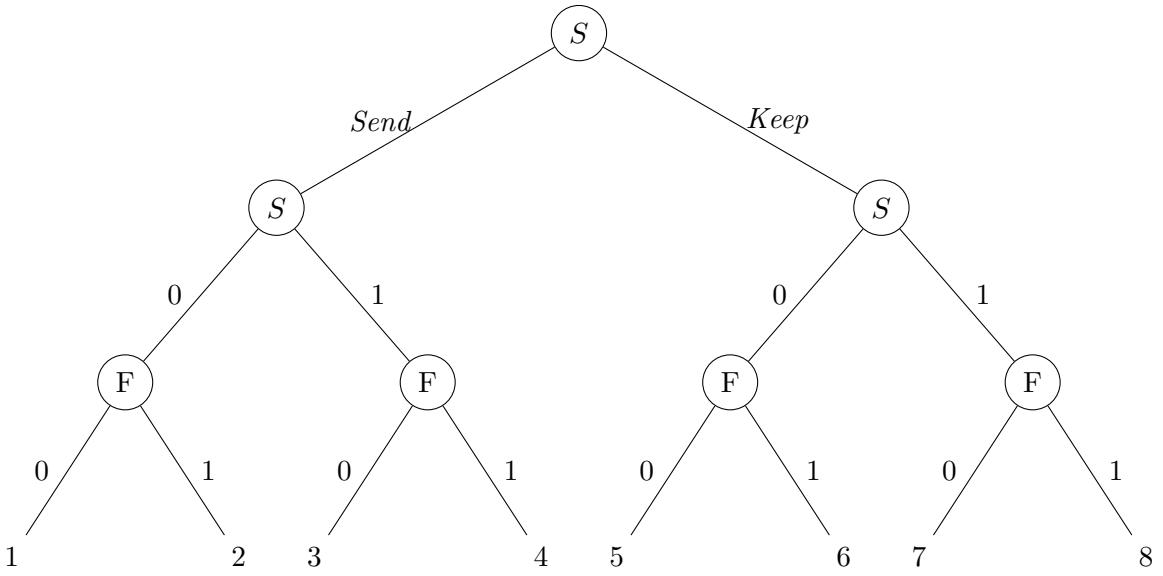
As the Buyer has to trust the Mediator, one possible approach is choosing a loyal friend of the Buyer to be the Mediator. It is important that he is loyal to the Buyer, meaning that he will not betray the Buyer and keep the physical good G to himself. The Buyer fully trusts his loyal friend.

The friend will not act rational, as he does not want to maximize his own Utility. He rather wants to maximize the Buyer's utility. Therefore he will always choose the option that maximizes the payout of the Buyer. The Friend(F) will receive the physical good G and check if it matches the agreed good. After that, he reports 0 (received the good successfully) or 1 (did not receive the good successfully) to the Smart Contract. After that he will send the G to the Buyer.

To increase readability, we do not plot the decisions from point 1 and point 2. If someone does not deposit the collateral, the Smart Contract aborts the trade and the payout is 0|0.

Two side collateral: Friend reports	
1 :	Seller deposits collateral $C_S > P$ to the Smart Contract
2 :	Buyer deposits collateral $C_B > P$ and price P to the Smart Contract
3 :	Seller sends physical good G to the Friend of the Buyer and reports it to the Smart Contract
4 :	Friend of the Buyer receives the physical good and reports it to the Smart Contract
5 :	The Smart Contract releases the collateral of the Seller and the Buyer and sends P to the Seller
6 :	Friend sends G to the Buyer

Figure 5.5: Two side collateral protocol: Friend of Buyer reports



	Payout
1	$-U_S(G) \mid U_B(G)$
2	$-U_S(G) - C_S \mid U_B(G) - P - C_B$
3	$P - U_S(G) - C_S \mid U_B(G) - P - C_B$
4	$P - U_S(G) \mid U_B(G) - P$
5	$0 \mid 0$
6	$-C_S \mid -P - C_B$
7	$-C_S \mid -P - C_S$
8	$P \mid -P$

We again start at the bottom of the tree to analyze the payout. As F wants to maximize the payout of the Buyer, he will choose:

- 1 over 2 ($U_B(G) > U_B(G) - P - C_B$)
- 4 over 3 ($U_B(G) - P > U_B(G) - P - C_B$)
- 5 over 6 ($0 > -P - C_B$)
- 8 over 7 ($-P > -P - C_B$))

The Seller would choose option 8, as

$$P > P - U_S(G) > 0 > -U_S(G)$$

Because the payout of the Buyer is $-P < 0$, he would not engage in this trade and therefore refuse to pay the collateral in step 1. The trade is still unsuccessful.

We modeled the friend as being loyal to the Buyer. He wants to maximize the utility of the Buyer. That means that he will always choose the behaviour that benefits the Buyer. In the case where the Seller does not send G but lies to the SC and reports 1, the friend of the Buyer will also report 1. In this crucial case he will behave exactly like the Buyer. That means that the game does not change for the Seller. It is still profitable for him to deviate from the protocol and not send G.

The friend is not suited as being the Mediator. He does not achieve the same functionality as the ideal Mediator proposed in definition 13, as he will not always report truthfully to the Smart Contract. As the Seller knows this, he can take advantage of it. He knows that he will not be punished if he does not send the physical good. The friend as Mediator is not a credible threat to the Seller, as he will report exactly like the Buyer would. Therefore a rational Seller will not send the good and this protocol does not achieve the goal proposed in section 3.2.1.

Introducing a friend as the Mediator does not solve the problem. The tree and the payout are exactly the same as in the two-sided collateral protocol (c.f. 4.15). To solve the problem we need to come up with a Mediator that is fully trusted by the Buyer, but is not trying to maximize the Buyer's payout. Every entity that maximizing the payout for the Buyer will not be a credible threat to the Seller.

5.4.2 Postal Office reports

In our formal analysis we assumed that there is a way for the physical item G to be reliably shipped. In practice physical goods are usually shipped by a another party (Post office/shipping company). Therefore the parties have to trust the Postal Office (PO) to actually deliver the good. Because the parties already trust the PO, the PO can also report to the Smart Contract and thereby act as the Mediator.

We propose the following protocol:

Two side collateral: Postal Office as Mediator	
1 :	Buyer gives SC address and specifications for G to the Postal Office
2 :	Seller deposits collateral $C_S > P$ to the Smart Contract
3 :	Buyer deposits collateral $C_B > P$ and price P to the Smart Contract
4 :	Seller sends physical good G to the Postal Office and reports it to the Smart Contract
5 :	Postal Office receives the physical good and reports it to the Smart Contract
6 :	The Smart Contract releases the collateral of the Seller and the Buyer and sends P to the Seller
7 :	Postal Office sends G to the Buyer

Figure 5.6: Two side collateral protocol: Postal Office as Mediator

Postal Office realises ideal Mediator

An honest Postal Office realises the ideal Mediator as defined in 6.1:

1. It receives the SC address and specifications that describe G
2. It receives the physical good from the Seller
3. It checks, if the received good matches the specifications for G
   - if so, it reports 1 and sends the good to the Buyer
   - else it reports 0 and destroys the good

As the Postal Office has to be trusted by the Buyer, it meets the requirements to being the ideal mediator. We have already shown (6.2) that a practical implementation of the ideal Mediator would solve our problem. Therefore the Postal Office reporting to the Smart Contract and thereby acting as Mediator is a viable solution to the problem.

Trust in the Postal Office

Although this solution with the PO as Mediator achieves an incoercible sale of the physical good G, both parties still have to trust the Postal Office behave honestly. In addition to that, the Postal Office now knows the specifications of the good being shipped. This means that there is less privacy for the Buyer and the Seller. If the physical good that is being shipped is very valuable, it opens up possible attacks by the Postal Office.

-po has to be big/trusted enough
-exit scams?

Difference Postal Office and Escrow

The big difference between the Postal Office and traditional third party escrow is that the PO does not handle the money. In traditional third party escrows, the escrow has to be trusted with the good and the money. As it is in possession of both, it opens up possibilities to scam the Buyer and the Seller out of both the money and the physical good. As the postal Office is only holding on to the physical good, the reward of a possible scam is considerably less than if it also had control of the money (approximately half, assuming that $U_{PO}(G) \approx P$).

We trust the PO by default? Third party escrow is third rational party therefore can scam, but PO is not modeled as rational party? kind of cheating

5.4.3 Machine reports

If we can build a machine that can recognize G reliably, we can completely eliminate trust. This machine can act as the Mediator. It does not make decisions, it simply takes inputs and acts accordingly. The inputs it takes are:

- The address of the SC
- Specifications that describe G
- The physical good G

The Mediator now is a deterministic machine in contrast to a third Entity. This has the advantage that both the Buyer and the Seller know that the Machine will behave exactly as it is programmed to do. There is no need of trust anymore. **[Orfeas: mention that the machine can presumably use image recognition, be cheap and open source, therefore both parties can verify its code and the seller can easily believe that the buyer can have such a machine, thus eliminating trust]**

Behaviour of the Machine

1. It receives the SC address and specifications that describe G
2. It receives the physical good from the Seller
3. It checks, if the received good matches the specifications for G
   - if so: it reports 1 and sends the good to the Buyer
   - else: it reports 0 and destroys the good

A Machine with this behaviour is realising the ideal Mediator from 6.1.

Protocol

Two side collateral: Machine as Mediator

1 : Buyer gives SC address and specifications for G to the Machine
2 : Seller deposits collateral $C_S > P$ to the Smart Contract
3 : Buyer deposits collateral $C_B > P$ and price P to the Smart Contract
4 : Seller sends physical good G to the Machine and reports it to the Smart Contract
5 : Machine receives the physical good and reports it to the Smart Contract
6 : The Smart Contract releases the collateral of the Seller and the Buyer and sends P to the Seller
7 : Machine sends G to the Buyer

Figure 5.7: Two side collateral protocol: Machine as Mediator

As the machine realises an ideal ideal Mediator (6.1), this protocol achieves an incoercible sale of G. In addition to that, no trust is needed as the Mediator is not able to behave differently. It will always behave as it was programmed to behave. Therefore it is not able to perform any kind of scam.

Realizability of the machine

The biggest challenge for the machine is to determine if the provided good actually matches the specifications.
We believe that a machine which behaves exactly as we described is could be realised using artificial intelligence. **[Orfeas: expand a bit. Mention image/3d object recognition. Find references]**

5.5 Difference from trusted third party/ Escrow

How is the reporting entity different than an escrow? -only buyer has to trust? -Maybe even model rational: misbehaving is doing more damage to the company/manufacturer than behaving exactly as described -¿ then no trust is necessary -¿ but then we can us it as a clean middleman that reports to the SC

6 Conclusion

[Nikos: still need to add references]
Cryptocurrencies gain acceptance as a valid payment method for digital and physical goods. There is a need for a protocol that handles these sales with a cryptocurrency as payment method. Current solutions use a trusted third party. As there are situations where no trusted third party is available and the third party can misbehave, protocols without the need of that trust are advantageous. The goal of this thesis was to find such a protocol that enables a trustless and incoercible sale of physical goods over a blockchain.

First, we formalized the problem. After that we proposed two-party protocols and analyzed if these protocols are viable solutions to the formalized problem. First we examined protocols without collateral. Then we looked at protocols where only one party deposits collateral. After showing that these approaches do not solve the proposed problem, we looked at protocols where both parties deposit collateral. As all of these candidates did not solve the proposed problem, we tried to prove that such a two-party protocol is impossible. **[Nikos: Here write something about the proof]** After showing that indeed such a two-party protocol is impossible, we lowered the requirements for the protocol by allowing a Mediator which is trusted by the Buyer. We showed that with this mediator there exists a protocol that enables a trustless and incoercible sale of physical goods. We proposed such a protocol with an idealised Mediator in section 5.3. At the end of the thesis, we looked at possible practical realizations that can take the role of the Mediator. We proposed two possible solutions (Postal Office reports, Machine reports).

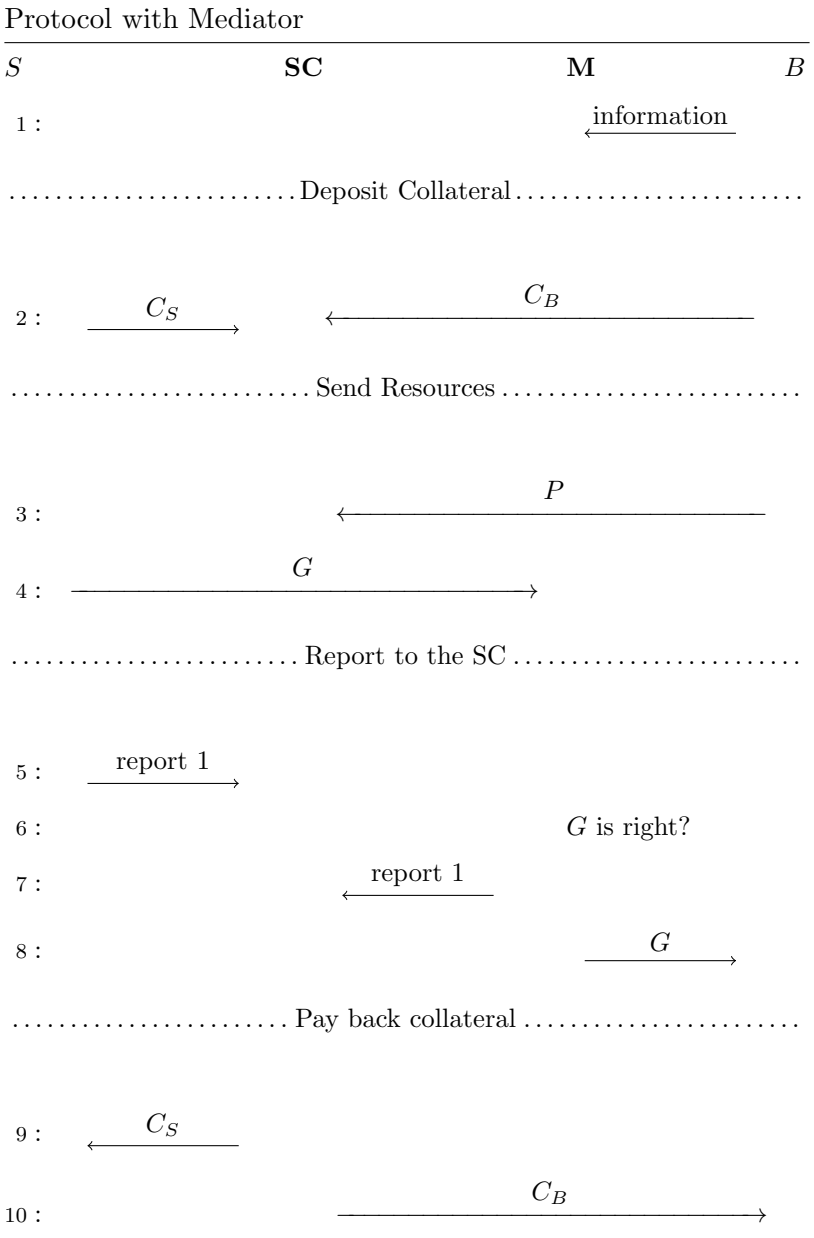
6.1 Outlook

maybe find a formal way to distinguish working protocols from not working ones, given the new assumptions where we allow a Mediator? probably dependend on the behaviour of the Mediator. why is it working? Is there an easier protocoll that still solves the problem?

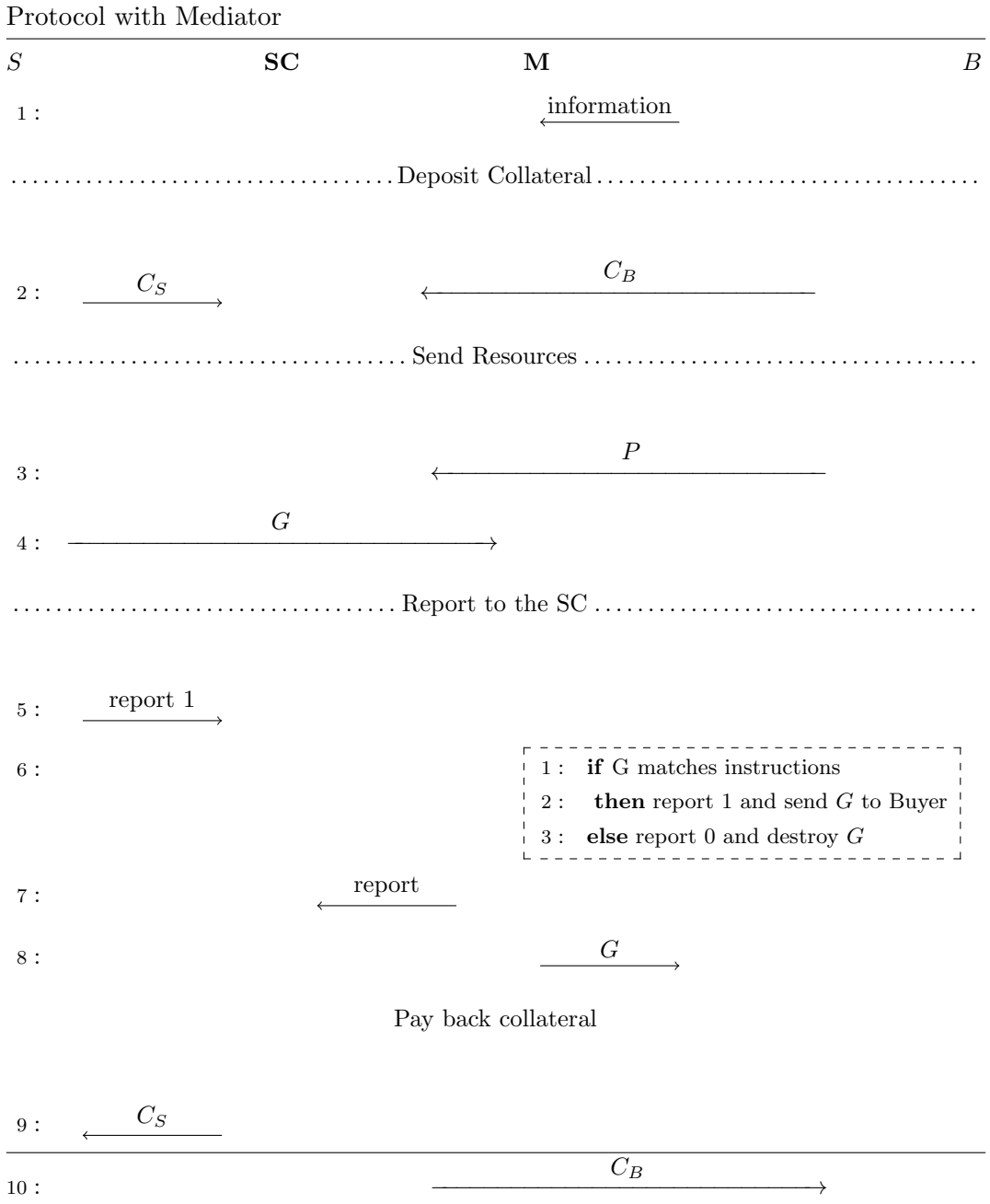
Engeneer the (ML) machine, as in section... that can classify a physical good correctly with high probability. Object detection with 3d model. maybe loook at face ID?

7 test

7.0.1 without subprocedure



7.0.2 with subprocedure

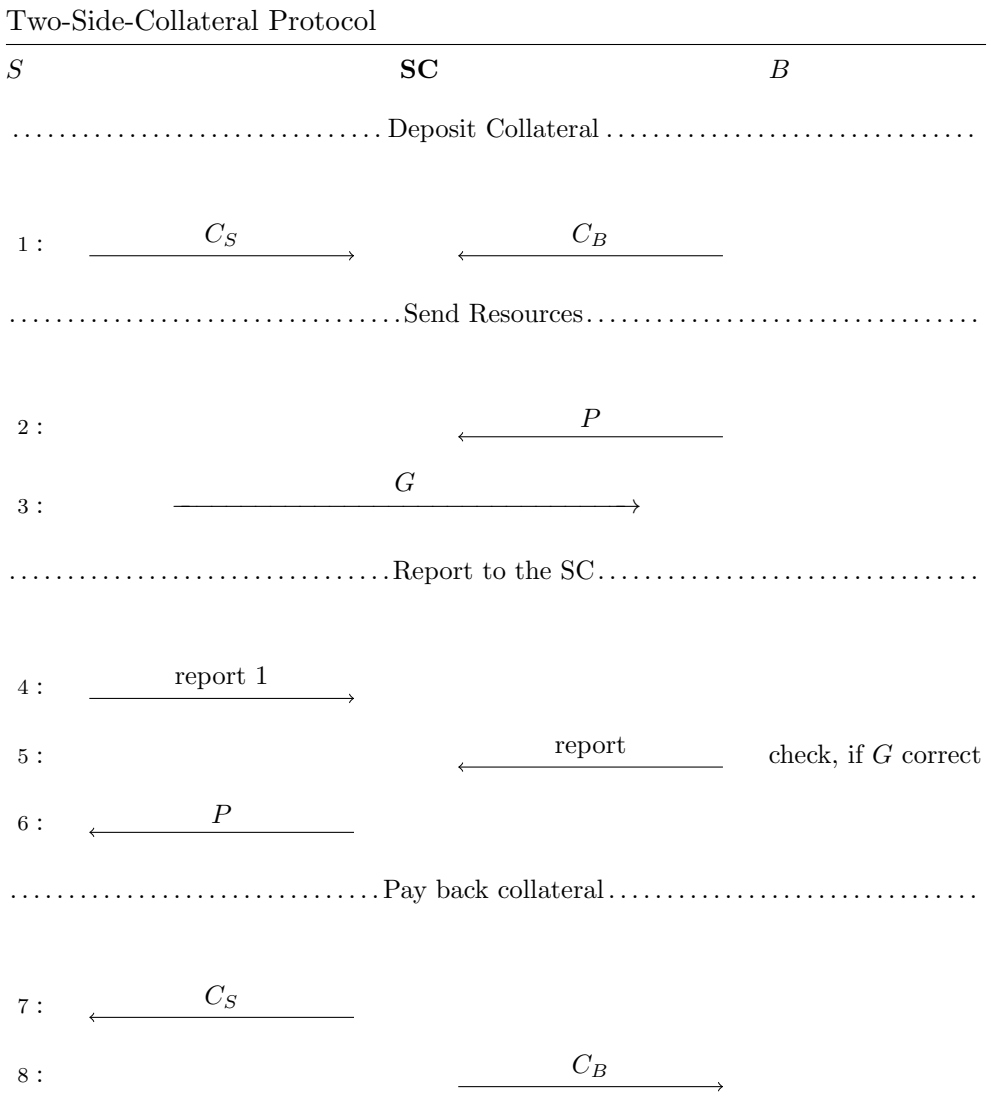


7.1 two-Side collateral

How it is now

1. Seller deposits collateral $C_S > P$ to the Smart Contract
2. Buyer deposits collateral $C_B > P$ to the Smart Contract
3. Buyer sends funds of amount P **[Orfeas: sends P coins]** to the Smart Contract **[Orfeas: this step could be combined with the previous one]**
4. Seller sends physical good G to the seller and reports it to the Smart Contract
5. Buyer receives the physical good and confirms **[Orfeas: reports]** it to the Smart Contract
6. The Smart Contract releases the collateral of the Seller and the Buyer and sends P to the Seller

cryptocode figure



cryptocode proocedure

Protocol

1 : Seller deposits collateral $C_S > P$ to the Smart Contract
2 : Buyer deposits collateral $C_B > P$ and P coins to the Smart Contract
3 : Seller sends physical good G to the seller and reports it to the Smart Contract
4 : Buyer receives the physical good and reports it to the Smart Contract
5 : The Smart Contract releases the collateral of the Seller and the Buyer and sends P to the Seller

Figure 7.1: Double collateral protocol

Bibliography

[AK19] Aditya Asgaonkar and Bhaskar Krishnamachari. "Solving the Buyer and Seller's Dilemma: A Dual-Deposit Escrow Smart Contract for Provably Cheat-Proof Delivery and Payment for a Digital Good without a Trusted Mediator". en. In: *2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*. Seoul, Korea (South): IEEE, May 2019, pp. 262–267. ISBN: 978-1-72811-328-9. DOI: 10.1109/BLOC.2019.8751482. URL: https://ieeexplore.ieee.org/document/8751482/ (visited on 05/23/2022).

[Big+15] Giancarlo Bigi et al. "Validation of Decentralised Smart Contracts Through Game Theory and Formal Methods". en. In: *Programming Languages with Applications to Biology and Security*. Ed. by Chiara Bodei, Gianluigi Ferrari, and Corrado Priami. Vol. 9465. Series Title: Lecture Notes in Computer Science. Cham: Springer International Publishing, 2015, pp. 142–161. ISBN: 978-3-319-25526-2 978-3-319-25527-9. DOI: 10.1007/978-3-319-25527-9_11. URL: http://link.springer.com/10.1007/978-3-319-25527-9_11 (visited on 05/23/2022).

[Gol+17] Steven Goldfeder et al. "Escrow Protocols for Cryptocurrencies: How to Buy Physical Goods Using Bitcoin". en. In: *Financial Cryptography and Data Security*. Ed. by Aggelos Kiayias. Vol. 10322. Series Title: Lecture Notes in Computer Science. Cham: Springer International Publishing, 2017, pp. 321–339. ISBN: 978-3-319-70971-0 978-3-319-70972-7. DOI: 10.1007/978-3-319-70972-7_18. URL: http://link.springer.com/10.1007/978-3-319-70972-7_18 (visited on 05/23/2022).

[LS08a] Kevin Leyton-Brown and Yoav Shoham. "Essentials of Game Theory: A Concise Multidisciplinary Introduction". en. In: *Synthesis Lectures on Artificial Intelligence and Machine Learning* 2.1 (Jan. 2008), pp. 1–88. ISSN: 1939-4608, 1939-4616. DOI: 10.2200/S00108ED1V01Y200802AIM003. URL: http://www.morganclaypool.com/doi/abs/10.2200/S00108ED1V01Y200802AIM003 (visited on 05/23/2022).

[LS08b] Kevin Leyton-Brown and Yoav Shoham. *Essentials of Game Theory: A Concise, Multidisciplinary Introduction*. en. Cham: Springer International Publishing, 2008. ISBN: 978-3-031-00417-9 978-3-031-01545-8. DOI: 10.1007/978-3-031-01545-8. URL: https://link.springer.com/10.1007/978-3-031-01545-8 (visited on 09/30/2022).

[Nak] Satoshi Nakamoto. "Bitcoin: A Peer-to-Peer Electronic Cash System". en. In: (), p. 11.

[NMR44] John von Neumann, Oskar Morgenstern, and Ariel Rubinstein. *Theory of Games and Economic Behavior (60th Anniversary Commemorative Edition)*. Princeton University Press, 1944. ISBN: 9780691130613. URL: http://www.jstor.org/stable/j.ctt1r2gkx (visited on 10/01/2022).

[Ras] Eric Rasmusen. "GAMES AND INFORMATION, FOURTH EDITION". en. In: (), p. 577.

[Zim] David Zimbeck. "Two Party double deposit trustless escrow in cryptographic networks and Bitcoin." en. In: (), p. 3.